SECTION D'INFORMATIQUE
DE L'ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ANNÉE ACADÉMIQUE 2017 - 2018

Le livret des cours est aussi disponible depuis l'adresse internet de la section :

https://ic.epfl.ch/informatique/archives-livrets-cours

Ordonnance
**sur la formation menant au bachelor et au master
de l'Ecole polytechnique fédérale de Lausanne
(Ordonnance sur la formation à l'EPFL)**

414.132.3

du 14 juin 2004 (Etat le 1er septembre 2016)

La Direction de l'Ecole polytechnique fédérale de Lausanne (EPFL),
vu l'art. 3, al. 1, let. b, de l'ordonnance du 13 novembre 2003 sur l'EPFZ
et l'EPFL[1],
arrête:

Section 1   Généralités et définitions

Art. 1   Objet

[1] La présente ordonnance régit la formation menant aux titres de bachelor et de master décernés par l'EPFL.

[2] Les études de bachelor et de master constituent les deux phases successives de cette formation.

Art. 2   Admission

L'admission à la formation menant au bachelor et au master est déterminée par l'ordonnance du 8 mai 1995 concernant l'admission à l'Ecole polytechnique fédérale de Lausanne[2].

Art. 3   Titres

[1] L'EPFL décerne les titres suivants dans ses domaines d'études (sections ou domaines):[3]

    a.   le bachelor;

    b.   le master.

[2] Les titres sont munis du sceau de l'EPFL et mentionnent le nom du titulaire. Ils sont signés par le président de l'EPFL, par le vice-président pour les affaires académiques à l'EPFL et par le directeur de section. Ils sont accompagnés du «diploma supplement» décrivant le niveau, le contexte, le contenu et le statut des études accomplies avec succès. Les titres mentionnent le domaine d'études et, pour le

master, la désignation professionnelle du titulaire, ainsi qu'une éventuelle orientation particulière.

[3] …[4]

[4] Tout titulaire du diplôme de l'EPFL (art. 15, al. 1) est autorisé à se présenter comme titulaire du master de l'EPFL (annexe I).

[5] La liste des titres et désignations correspondantes selon les domaines d'études figure dans l'annexe I de la présente ordonnance.

[6] Les titres de master décernés par l'EPFL communément avec d'autres institutions sont régis par les accords spécifiques.

[7] L'EPFL décerne également le titre de docteur ès sciences (ou Ph. D.) et d'autres titres correspondant à la formation continue. Ces titres font l'objet d'ordonnances spécifiques.

Art. 4 Crédits d'études ECTS

[1] L'EPFL attribue des crédits pour les prestations d'études contrôlées, conformément au système européen de transfert et d'accumulation de crédits d'études (European Credit Transfer and Accumulation System, ci-après ECTS). Le nombre de crédits défini pour une matière est fonction du volume de travail à fournir pour atteindre l'objectif de formation.

[2] Les crédits ECTS sont acquis de façon cumulative selon les conditions définies par l'ordonnance du 30 juin 2015 sur le contrôle des études à l'EPFL[5]. Les règlements d'application du contrôle des études visés à l'art. 5 de ladite ordonnance définissent le nombre de crédits attribué à chaque domaine d'études.[6]

[3] Les plans d'études visés à l'art. 5 de l'ordonnance sur le contrôle des études à l'EPFL sont conçus de façon à permettre l'acquisition de 60 crédits ECTS par année académique.[7]

Art. 5[8] Nombre de crédits ECTS requis

[1] A réussi le bachelor l'étudiant qui a acquis 180 crédits ECTS conformément à l'ordonnance du 30 juin 2015 sur le contrôle des études à l'EPFL[9] et aux règlements d'application visés à l'art. 5 de ladite ordonnance.

[2] A réussi le master l'étudiant qui a acquis, en sus du bachelor, 60 crédits ECTS, respectivement 90 crédits ECTS pour les sections qui les requièrent conformément à

---

[4] Abrogé par le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, avec effet au 1er sept. 2016 (RO **2015** 2539).

[5] RS **414.132.2**

[6] Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

[7] Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

[8] Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

[9] RS **414.132.2**

l'annexe I, et réussi le projet de master représentant 30 crédits, conformément à l'ordonnance sur le contrôle des études à l'EPFL et aux règlements d'application.

Section 2   Bachelor

Art. 6   Etapes de formation

[1] Le bachelor de l'EPFL est composé de deux étapes successives de formation:

    a.   le cycle propédeutique;

    b.   le cycle bachelor.

[2] …[10]

Art. 7   Cycle propédeutique

[1] Le cycle propédeutique s'étend sur deux semestres.[11]

[2] Il a pour objectif la vérification des connaissances de base, l'acquisition des compétences nécessaires pour la suite de la formation en sciences naturelles et une initiation dans les sciences humaines et sociales.

[3] Sa durée ne peut excéder deux ans.

[4] La réussite de l'examen propédeutique permet d'acquérir 60 crédits ECTS et est la condition pour entrer au cycle bachelor.

Art. 8   Cycle bachelor

[1] Le cycle bachelor s'étend sur deux années d'études.

[2] Il a pour objectif l'acquisition des bases scientifiques générales et spécifiques au domaine d'études et à un secteur des sciences humaines et sociales.

[3] Il doit être réussi au plus tard quatre ans après la réussite du cycle propédeutique ou, en cas d'admission à un semestre supérieur, dans un délai qui correspond au double du nombre de semestres à accomplir.[12]

[4] Le cycle bachelor est réputé réussi par l'acquisition de 120 crédits ECTS. La réussite du cycle bachelor est la condition pour entrer au cycle master. L'art. 29, al. 1, de l'ordonnance du 30 juin 2015 sur le contrôle des études à l'EPFL[13] est réservé.[14]

---

[10]   Abrogé par le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, avec effet au 1er sept. 2016 (RO **2015** 2539).

[11]   Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

[12]   Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

[13]   RS **414.132.2**

[14]   Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

Section 3   Master

Art. 9   Etapes de formation

[1] Le master est composé de deux étapes successives de formation:

    a.   le cycle master;

    b.   le projet de master.

[2] …[15]

Art. 10   Cycle master

[1] Il a pour objectif l'acquisition des connaissances spécifiques du domaine d'études permettant la maîtrise de la profession, ainsi que l'étude d'une discipline des sciences humaines et sociales.

[2] La durée du cycle master de 60 crédits ECTS est d'une année, mais ne peut excéder deux ans; celle du cycle de 90 crédits ECTS est d'une année et demie, mais ne peut excéder trois ans.

[3] Le cycle master est réputé réussi par l'acquisition de 60 ou 90 crédits ECTS.

Art. 11[16]   Projet de master

[1] Le projet de master s'étend sur un semestre et sa réussite permet d'acquérir 30 crédits ECTS.

[2] Le projet de master doit être réussi dans le délai d'un an après la réussite du cycle master ou, le cas échéant, après l'admission conditionnelle (art. 29, al. 3, de l'ordonnance du 30 juin 2015 sur le contrôle des études à l'EPFL[17]).

[3] La réussite du cycle master est la condition pour entamer le projet de master. L'art. 29, al. 3, de l'ordonnance du 30 juin 2015 sur le contrôle des études à l'EPFL est réservé; s'il s'applique, la réussite du projet de master implique la réussite préalable du cycle master.

Section 4   Durées de formation

Art. 12[18]   Conditions liées aux durées

[1] Les crédits requis doivent être acquis dans les durées fixées pour chaque cycle de formation par la présente ordonnance.

---

[15]   Abrogé par le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, avec effet au 1er sept. 2016 (RO **2015** 2539).

[16]   Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

[17]   RS **414.132.2**

[18]   Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

² En dérogation à l'al. 1, l'école peut prolonger la durée maximale d'un cycle de formation ou accorder une interruption entre deux cycles à un étudiant qui fait valoir un motif valable, notamment une longue maladie, une maternité, une période d'obligation de servir, dès qu'il en a connaissance et avant l'échéance de la durée maximale.

Section 5   Autres modalités

Art. 13   Mobilité

¹ Au titre de la mobilité, l'EPFL peut autoriser les étudiants à étudier un semestre ou un an dans une autre haute école, ou à faire le projet de master dans une autre haute école, dans le secteur public ou dans l'industrie, en restant immatriculés à l'EPFL. Les contrôles des acquis passés avec succès dans une autre haute école sont pris en compte pour autant que le programme d'études ait été préalablement fixé avec le responsable du domaine d'études de l'EPFL.

² Les directives de l'école s'appliquent.[19]

Art. 14   Modification du droit en vigueur

La modification du droit en vigueur est réglée dans l'annexe II.

Art. 15   Dispositions transitoires

¹ Le diplôme est décerné jusqu'au 31 décembre 2004.

² Les titres de bachelor et de master sont décernés à partir du 1er janvier 2005.

Art. 16   Entrée en vigueur

¹ La présente ordonnance entre en vigueur le 18 octobre 2004, à l'exception de l'al. 2.

² L'annexe II entre en vigueur le 1er janvier 2005.

---

[19]   Nouvelle teneur selon le ch. I de l'O de la Direction de l'EPFL du 30 juin 2015, en vigueur depuis le 1er sept. 2016 (RO **2015** 2539).

Ordonnance
414.132.2

sur le contrôle des études menant au bachelor et au master à l'Ecole polytechnique fédérale de Lausanne
(Ordonnance sur le contrôle des études à l'EPFL)

du 30 juin 2015 (Etat le 1er septembre 2016)

La Direction de l'Ecole polytechnique fédérale de Lausanne (Direction de l'EPFL),
vu l'art. 3, al. 1, let. b, de l'ordonnance du 13 novembre 2003 sur l'EPFZ
et l'EPFL[1],
arrête:

Chapitre 1 Dispositions générales
Section 1 Objet et champ d'application

Art. 1   Objet

La présente ordonnance arrête les règles de base du contrôle des études à l'EPFL.

Art. 2   Champ d'application

[1] La présente ordonnance s'applique à la formation menant au bachelor et au master de l'EPFL.

[2] Dans la mesure où la direction de l'EPFL n'a pas édicté de règles particulières, les art. 8, 10, 12, 14, 15 et 18 à 20 s'appliquent également:

    a.   aux examens d'admission;

    b.   aux examens du cours de mathématiques spéciales (CMS);

    c.   aux examens du cours de mise à niveau;

    d.   aux examens de doctorat;

    e.   aux examens des programmes doctoraux;

    f.   aux examens de la formation continue et de la formation approfondie.

Section 2 Définitions générales

Art. 3   Branche

[1] Une branche est une matière d'enseignement faisant l'objet d'une ou de plusieurs épreuves.

[2] Une branche dite de semestre est une branche dont les épreuves se déroulent pendant la période de cours.

[3] Une branche dite de session est une branche dont une épreuve se déroule en session d'examens. Elle peut comporter des épreuves se déroulant pendant la période de cours.

[4] Une branche de semestre peut consister en un stage.

Art. 4 Crédits et coefficients

A toute branche est associé un nombre de crédits ECTS (European Credit Transfer and Accumulation System) (crédits) ou, pour le cycle propédeutique, un coefficient, qui indiquent son poids dans la formation.

Section 3
Dispositions communes aux études de bachelor et de master

Art. 5 Plans d'études et règlements d'application

Des plans d'études et des règlements d'application sont édictés pour chaque cycle d'études de chaque domaine. Ils définissent en particulier:

a. les branches de semestre et les branches de session;

b. le semestre ou la session pendant lesquels ces branches peuvent être présentées;

c. la forme (écrite ou orale) de l'épreuve en session;

d. la composition des blocs et des groupes de branches;

e. les coefficients ou les crédits attribués à chaque branche;

f. le nombre de crédits ou le coefficient à acquérir dans chaque bloc et chaque groupe;

g. les conditions applicables aux prérequis (art. 25);

h. les conditions de réussite particulières;

i. les études d'approfondissement, de spécialisation ou interdisciplinaires;

j. les éventuels régimes transitoires applicables aux modifications des plans d'études et des règlements d'application.

Art. 6 Blocs et groupes de branches

[1] Les branches sont rassemblées en bloc ou en groupe. Chaque branche ne peut faire partie que d'un seul bloc ou d'un seul groupe. Un bloc peut être constitué d'une seule branche.

² Un bloc est réputé réussi:

  a. lorsque la somme des crédits acquis par branche est égale ou supérieure au nombre requis; ou

  b. lorsque la somme des crédits acquis pour les branches présentées atteint le nombre requis et que la moyenne du bloc (art. 8, al. 5) est égale ou supérieure à 4,00; dans ce cas, la totalité des crédits des branches présentées est acquise.

³ Un groupe est réputé réussi lorsque les crédits des branches qui le composent ont été accumulés jusqu'au nombre requis; aucune compensation n'est possible entre les notes des branches du groupe.

Art. 7 Fiches de cours

Les fiches de cours publiées indiquent en particulier, pour chaque branche:

  a. les objectifs de formation;

  b. un bref descriptif de la matière;

  c. les épreuves composant la note finale, avec leur pondération et leur forme;

  d. les éventuels prérequis (art. 25);

  e. la langue d'enseignement.

Art. 8 Notation

¹ Une épreuve est notée de 1,00 à 6,00. Les notes en dessous de 4,00 sanctionnent des prestations insuffisantes. L'épreuve est notée 0 lorsque l'étudiant ne se présente pas, ne répond à aucune question ou ne respecte pas les délais.

² La note finale de la branche se compose des notes de ses épreuves. Elle est arrêtée au quart de point. Lorsqu'elle est inférieure à 1,00, la branche est considérée comme non acquise et notée NA (non acquis). L'appréciation NA compte comme tentative de réussite.

³ Le règlement d'application peut prévoir qu'une branche est notée au moyen des appréciations R (réussi) ou E (échec).

⁴ Lorsque la branche est répétée, la note retenue est celle de la seconde tentative.

⁵ Les moyennes sont calculées en pondérant chaque note finale chiffrée de branche par son coefficient ou son nombre de crédits. Elles sont arrêtées au centième. Les appréciations NA et E empêchent l'obtention d'une moyenne, sauf dans les cas visés à l'art. 6, al. 2, let. b, et 3.

Art. 9 Organisation des sessions et des épreuves et inscriptions aux branches

¹ Deux sessions d'examens sont organisées par année académique. Elles ont lieu entre les semestres.

2 Les délais d'inscription aux branches, les délais de retrait, les horaires et les dates des épreuves, ainsi que les autres modalités sont communiqués aux étudiants.

3 A l'échéance des délais, les inscriptions aux branches et les retraits sont définitifs.

4 Lorsque l'étudiant répète une branche, celle-ci est régie par les dispositions en vigueur au moment de la répétition, à moins que l'école n'en ait disposé autrement.

Art. 10   Incapacité

1 L'étudiant qui se prévaut d'un motif d'incapacité à se présenter à une épreuve doit l'annoncer à l'école dès la survenance de ce motif.

2 Il lui présente en outre les pièces justificatives au plus tard trois jours après la survenance du motif d'incapacité. Par pièces justificatives, on entend notamment un certificat médical ou une attestation d'une obligation légale de servir.

3 Invoquer un motif d'incapacité après s'être présenté à l'épreuve ne justifie pas l'annulation d'une note.

Art. 11   Langue des épreuves

1 Les épreuves se déroulent dans la langue de l'enseignement de la branche.

2 L'étudiant a le droit de répondre en français à une épreuve en anglais. Sur demande écrite de sa part, l'enseignant peut lui accorder de répondre en anglais si l'épreuve est en français.

Art. 12   Etudiants en situation de handicap

1 Si un candidat en situation de handicap en fait la demande au début de l'année académique, l'école fixe un déroulement d'épreuve adapté à son handicap et décide de l'utilisation de moyens auxiliaires ou de l'assistance personnelle nécessaires.

2 Les objectifs de l'épreuve doivent être garantis.

Art. 13   Tâches de l'enseignant

1 L'enseignant remplit notamment les tâches suivantes:

   a. donner les informations nécessaires sur ses matières d'enseignement pour qu'elles soient publiées dans la fiche de cours;

   b. informer les étudiants, s'il y a lieu, du contenu des matières et du déroulement des épreuves;

   c. conduire les épreuves;

   d. prendre des notes de chaque épreuve orale, qu'il peut être appelé à produire auprès de la conférence d'examen ou des autorités de recours;

   e. attribuer les notes des épreuves, ainsi que la note finale de branche;

f.   conserver pendant six mois après la fin du cycle concerné (chap. 2 à 4) les épreuves écrites et les notes prises durant les épreuves orales; en cas de recours, ce délai est prolongé jusqu'au terme de la procédure.

2 S'il est empêché de remplir ses tâches, le directeur de section désigne un remplaçant.

Art. 14   Observateur

1 Un observateur désigné par le directeur de section assiste à l'épreuve orale ayant lieu en session d'examens, dans le but de veiller à son déroulement régulier.

2 Il prend, pour chaque candidat, des notes sur le déroulement de l'épreuve et les conserve conformément à l'art. 13, al. 1, let. f.

Art. 15   Consultation des épreuves

L'étudiant peut consulter son épreuve dans les 6 mois qui suivent la communication du résultat.

Art. 16   Commissions d'évaluation

Des commissions d'évaluation peuvent être mises sur pied pour les branches de semestre. Outre l'enseignant et un expert, les commissions d'évaluation peuvent comprendre les assistants et les chargés de cours qui ont participé à l'enseignement, ainsi que d'autres professeurs.

Art. 17   Conférence d'examen

1 La conférence d'examen siège à l'issue de chaque session. Elle est composée du vice-provost pour la formation, qui la préside, du directeur de section et du chef du service académique. Les membres de la conférence d'examen peuvent se faire représenter par leur suppléant.

2 La conférence d'examen se prononce sur les cas particuliers conformément aux dispositions légales.

Art. 18   Fraude

1 Par fraude, on entend toute forme de tricherie en vue d'obtenir pour soi-même ou pour autrui une évaluation non méritée.

2 En cas de fraude, de participation à la fraude ou de tentative de fraude, le règlement disciplinaire du 15 décembre 2008 concernant les étudiants de l'Ecole polytechnique fédérale de Lausanne[2] s'applique.

---

2   RS **414.138.2**

Art. 19   Notification des résultats et communications

¹ La décision de réussite ou d'échec pour le cycle d'études est notifiée à l'étudiant.

² Elle fait mention des notes obtenues et des crédits acquis.

³ La notification de la décision ainsi que les communications ont lieu par voie électronique ou postale.

Art. 20   Demande de nouvelle appréciation et recours administratif

¹ La décision peut faire l'objet d'une demande de nouvelle appréciation auprès de l'école dans les 10 jours qui suivent sa notification. L'art. 63, al. 1, 3 et 4, de la loi fédérale du 20 décembre 1968 sur la procédure administrative³ est applicable.

² Elle peut également faire l'objet d'un recours administratif auprès de la commission de recours interne des EPF, dans les 30 jours qui suivent sa notification.

Chapitre 2   Examens du cycle propédeutique

Art. 21   Conditions de réussite

¹ L'étudiant qui, à l'issue du premier semestre du cycle propédeutique et de la session d'examens afférente, a atteint une moyenne pondérée (art. 8, al. 5) d'au moins 3,50 pour le premier bloc au sens du règlement d'application est admis au second semestre du cycle.

² A réussi le cycle propédeutique l'étudiant qui, conformément au plan d'études et au règlement d'application:

   a.   a présenté toutes les branches; et

   b.   a obtenu une moyenne égale ou supérieure à 4,00 dans chacun des blocs et, le cas échéant, les coefficients requis dans un groupe.

Art. 22   Echec et élimination

¹ Constituent un échec, au niveau du cycle propédeutique:

   a.   la non-atteinte d'une moyenne pondérée d'au moins 3,50 pour le premier bloc, à l'issue du premier semestre et de la session d'examens afférente;

   b.   la non-atteinte d'une moyenne pondérée d'au moins 4,00 par bloc ou la non-atteinte du nombre de coefficients requis dans un groupe, à l'issue du cycle propédeutique; ou

   c.   le fait de ne pas avoir présenté toutes les branches du cycle propédeutique, sous réserve de l'art. 23, al. 4.

³   RS **172.021**

² L'étudiant qui suit le cycle propédeutique en première tentative et se trouve dans la situation visée à l'al. 1, let. a, suit au second semestre le cours de mise à niveau de l'EPFL.

³ Est assimilé à un échec au cycle propédeutique de l'EPFL un échec ou une absence de réussite subi dans une autre haute école à un niveau comparable au cycle propédeutique, si la majorité des branches sont considérées par l'EPFL comme étant analogues.

⁴ Constitue un échec définitif un second échec au niveau du cycle propédeutique ou le non-respect de la durée maximale de deux ans pour réussir le cycle.

⁵ Constituent un motif d'exclusion définitive de toute formation de bachelor à l'EPFL la non-atteinte d'une moyenne pondérée d'au moins 4,00 à l'issue du cours de mise à niveau ou le non-respect de l'obligation de le suivre.

Art. 23   Répétition

¹ L'étudiant qui est en situation d'échec, en première tentative, selon l'art. 22, al. 1, let. b et c, ou qui a atteint une moyenne d'au moins 4,00 au cours de mise à niveau est admis une seconde fois au premier semestre du cycle propédeutique de l'année académique qui suit.

² Les branches d'un bloc ou d'un groupe réussis (art. 21, al. 2, let. b) sont acquises et ne peuvent pas être répétées.

³ La répétition des autres branches non réussies est impérative. La répétition des branches réussies est facultative, sauf pour les étudiants issus de la situation visée à l'art. 22, al. 1, let. a, pour lesquels elle est obligatoire. Le règlement d'application peut toutefois prévoir que certaines branches de semestre réussies ne peuvent pas être répétées.

⁴ En cas d'absence justifiée au sens de l'art. 10, l'école examine s'il est raisonnablement exigible de l'étudiant qu'il complète le cycle propédeutique à la session ordinaire correspondante de l'année suivante ou si l'étudiant doit être considéré comme ayant échoué.

Chapitre 3   Examens du cycle bachelor et du cycle master

Art. 24   Crédits

Les crédits de la branche sont attribués lorsque la note obtenue est égale ou supérieure à 4,00 ou que la moyenne du bloc de branches à laquelle elle appartient est égale ou supérieure à 4,00.

Art. 25   Prérequis

Le règlement d'application ou la fiche de cours définit les branches dont l'étudiant doit avoir acquis les crédits afin d'être admis à suivre d'autres branches.

Art. 26 Conditions de réussite

[1] Les crédits requis du cycle bachelor et du cycle master doivent être acquis conformément à la présente ordonnance, à l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[4] et au règlement d'application.

[2] Dans le cycle bachelor, 60 crédits au moins doivent être acquis par tranche de deux ans.

Art. 27 Répétition

[1] Si, dans un bloc ou un groupe, le nombre de crédits requis n'est pas acquis, les branches dont la note est inférieure à 4,00 peuvent être répétées une fois, impérativement à la session ordinaire de l'année qui suit.

[2] L'étudiant qui échoue deux fois à une branche optionnelle peut en présenter une nouvelle conformément au plan d'études.

Art. 28 Echec définitif

Si l'étudiant n'acquiert pas les crédits requis conformément à la présente ordonnance et au règlement d'application, dans le respect des durées maximales fixées par l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[5], il se trouve en situation d'échec définitif.

Art. 29 Admission conditionnelle au cycle consécutif

[1] Peut être admis conditionnellement au cycle master consécutif l'étudiant qui:

    a.   n'a pas plus de 10 crédits manquants sur ceux requis par le plan d'études de dernière année du cycle bachelor de l'EPFL; et

    b.   n'est pas en situation d'échec définitif.

[2] L'étudiant admis conditionnellement au cycle master consécutif a l'obligation d'acquérir les crédits manquants du bachelor dans l'année de son admission conditionnelle, sous peine d'être exclu du cycle.

[3] Peut être admis conditionnellement au projet de master l'étudiant qui:

    a.   n'a pas plus de 8 crédits manquants sur ceux requis pour le cycle master y compris les études visées à l'art. 5, let. i;

    b.   n'est pas en situation d'échec définitif.

---

4   RS **414.132.3**
5   RS **414.132.3**

Chapitre 4 Projet de master

Art. 30 Déroulement

[1] Le sujet du projet de master est fixé ou approuvé par le professeur ou le maître d'enseignement et de recherche qui en assume la direction.

[2] Sur demande, le directeur de section peut confier la direction du projet de master à un professeur ou un maître d'enseignement et de recherche rattaché à une autre section ou à un collaborateur scientifique.

[3] L'examen du projet de master consiste en une évaluation de sa présentation finale suivie d'une interrogation orale devant l'enseignant qui a dirigé le projet et un expert externe à l'EPFL désigné par l'enseignant en accord avec le directeur de section. Seul l'enseignant peut inviter d'autres personnes à l'interrogation orale; celles-ci ne participent pas à l'évaluation.

[4] Si la qualité rédactionnelle du projet est jugée insuffisante, l'enseignant peut exiger que l'étudiant y remédie dans un délai de deux semaines à compter de l'interrogation orale.

Art. 31 Conditions de réussite

[1] Le projet de master est réputé réussi lorsque la note attribuée est égale ou supérieure à 4,00.

[2] Si le règlement d'application prévoit un stage associé au projet de master, celui-ci doit avoir été réussi préalablement.

Art. 32 Répétition

[1] En cas d'échec, un nouveau projet de master peut être présenté dans le respect de la durée maximale prévue par l'ordonnance du 14 juin 2004 sur la formation à l'EPFL[6].

[2] Un second échec constitue un échec définitif.

Chapitre 5 Dispositions finales

Art. 33 Abrogation

L'ordonnance du 14 juin 2004 sur le contrôle des études à l'EPFL[7] est abrogée.

---

[6] RS **414.132.3**
[7] [RO **2004** 4323, **2006** 4125, **2008** 3721]

Art. 34 Disposition transitoire

Le chapitre 2 de l'ordonnance du 14 juin 2004 sur le contrôle des études à l'EPFL[8] demeure applicable jusqu'au 31 août 2017 aux étudiants répétant le cycle propédeutique durant l'année académique 2016–2017.

Art. 35 Entrée en vigueur

La présente ordonnance entre en vigueur le 1er septembre 2016.

---

[8] RO **2004** 4323, **2006** 4125, **2008** 3721

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

PLAN D'ÉTUDES

INFORMATIQUE

2 0 17 - 2 0 18

arrêté par la direction de l'EPFL le 22 mai 2017

Directeur de la section	**Prof. P. Fua**
Conseillers d'études :	
Année propédeutique	**Prof. B. Faltings**
1ère année cycle bachelor	**Prof. P. Ienne**
2ème année cycle bachelor	**Prof. M. Odersky**
Cycle master	**Prof. Ch. Koch**
Projet de master	**Prof. G. Candea**
Responsable passerelle HES	**Mme S. Dal Mas**
Coordinatrice des stages d'ingénieur	**Mme S. Dal Mas**
Délégué à la mobilité	**M. J.-L. Benz**
Adjointe du directeur de section	**Mme S. Dal Mas**
Secrétaire Bachelor	**Mme C. Bigler**
Secrétaire Master	**Mme A. Veltro**

Aux cycles bachelor et master, selon les besoins pédagogiques, les heures d'exercices mentionnées dans le plan d'études pourront être intégrées dans les heures de cours ; les scolarités indiquées représentent les nombres moyens d'heures de cours et d'exercices hebdomadaires sur le semestre

2017-2018 **INFORMATIQUE** **Cycle propédeutique**
Cursus commun IN-SC

Code	Matières	Type de branches	Enseignants sous réserve de modification	Sections	Semestres BA1 c	e	p	BA2 c	e	p	Coeff.	Période des épreuves *	Type examen *
	Bloc 1 :										37		
CS-101	Advanced information, computation, communication I	Spécifique	Lenstra	SC	4	2					7	H	écrit
COM-102	Advanced information, computation, communication II	Spécifique	Rimoldi	SC				4	2		7	E	écrit
MATH-111e	Algèbre linéaire (en français) ou		Shokrollahi	MA									
MATH-111en	Algèbre linéaire (en anglais)	Polytechnique	Pach	MA	4	2					6	H	écrit
MATH-111pi	Algèbre linéaire (classe pilote)		Deparis	MA									
MATH-101e	Analyse I (en français) ou		Lachowska	MA									
MATH-101de	Analyse I (en allemand) ou	Polytechnique	Semmler	MA	4	2					6	H	écrit
MATH-101en	Analyse I (en anglais)		Patakfalvi	MA									
MATH-106e	Analyse II (en français) ou	Polytechnique	Lachowska	MA				4	2		6	E	écrit
MATH-106en	Analyse II (en anglais)		Mountford	MA									
PHYS-101c	Physique générale I (en français) ou		Fuerbringer	PH									
PHYS-101de	Physique générale I (en allemand) ou	Polytechnique	Gruetter	PH	3	2					5	H	écrit
PHYS-101en	Physique générale I (en anglais)		Bréchet	PH									
	Bloc 2 :										24		
CS-173	Conception de systèmes numériques	Spécifique	Kluter	IN				4		2	6	sem P	
HUM-1nn	Enjeux mondiaux	Polytechnique	Divers enseignants	CDH				2		2	4	sem P	
CS-107	Introduction à la programmation	Polytechnique	Sam	IN	2	3					5	sem A	
CS-108	Pratique de la programmation orientée-objet	Spécifique	Schinz	IN				2	2	6	9	sem P	
	Totaux :				17	11	0	16	6	10	61		
	Totaux par semaine :				28			32					

Remarques :

* Se référer à l'art. 3 al. 4 du règlement d'application

Les cours en allemand et en anglais sont disponibles sous réserve de la compatiblité des horaire des cours.

2017-2018 **INFORMATIQUE** — **Cycle Bachelor**

Code	Matières	Enseignants sous réserve de modification	Sections	Filières	BA3 c	BA3 e	BA3 p	BA4 c	BA4 e	BA4 p	BA5 c	BA5 e	BA5 p	BA6 c	BA6 e	BA6 p	Crédits 2e	Crédits 3e	Période des épreuves *	Type examen *
	Bloc A																14			
CS-210	Functional programming	Kuncak/Odersky	IN	1	2	2											5		sem A	
CS-206	Parallelism and concurrency	Kuncak/Odersky	IN	1				1	1	2							4		sem P	
CS-207	Programmation orientée système	Chappelier	IN	1				1	2								3		sem P	
CS-207a	Projet programmation système	Chappelier/Figueiredo	IN	1						2							2		sem P	
	Bloc B																	17			
CS-209	Architecture des systems-on-chip	Ienne	IN	2				2		2							4		sem P	
CS-208	Computer architecture	Stojilovic	IN	2				2		2							4		sem A	
COM-208	Computer networks	Argyraki	SC					2	2								5		sem A	
PHYS-114	General physics II	Dil	PH					2	2								4		H	écrit
	Bloc C																	20			
CS-250	Algorithms	Kapralov	IN	3				4	2								6		H	écrit
MATH-203c	Analyse III	Nguyen H-M	MA		2	2											4		H	écrit
MATH-232	Probabilités et statistique	Thibaud	MA					4	2								6		E	écrit
CS-251	Theory of computation	Vishnoi	IN	3				2	2								4		sem P	
	Bloc D																		13		
CS-322	Introduction to database systems	Ailamaki /Koch	IN	1										2	1	1	4		E	écrit
CS-323	Introduction to operating systems	Zwaenepoel	IN	1										2	1		3		sem P	
CS-305	Software engineering	Candea	IN	1							2		3				6		sem A	
	Bloc E																		7		
CS-307	Introduction to multiprocessor architecture	Falsafi	IN	2							2		1				3		sem A	
COM-301	Sécurité des réseaux	Oechslin	SC								2	1					4		H	écrit
	Groupe "projet"																		8		
CS-398	Projet en Informatique I	Divers enseignants	IN								←	2	→					8	sem A ou P	
	Groupe "options"								←	33	→							5	28		
MATH-310	Algèbre	Testermann	MA								2	1						3	H	écrit
MATH-207d	Analyse IV	Nguyen H-M	MA	5	8			2	2								4		E	écrit
CS-308	Calcul quantique	Macris	SC											3	1		4		E	écrit
CH-361	Chimie générale avancée (selon disp. horaire)	Divers enseignants	CGC	7							3	2						6	H	écrit
EE-204	Circuits et systèmes I	Rachidi	EL	5			2	1									3		H	écrit
EE-205	Circuits et systèmes II	Gastpar	SC	4 5	8		2	1									4		E	écrit
CS-320	Computer language processing	Kuncak	IN								2	2	2					6	sem A	
EE-200	Électromagnétisme I : lignes et ondes	Fleury	EL				2	1									3		H	écrit
EE-201	Électromagnétisme II : calcul des champs	Fleury	EL										2	1			3		E	écrit
EE-202b	Électronique I	Zysman	SC				2	1									4		sem A	
EE-203b	Électronique II	Zysman	SC							2	2						4		sem A	
EE-381	Électronique III	Zysman	SC										2	1			3		sem P	
CS-321	Informatique du temps réel	Decotignie	SC				3		1								4		H	écrit
CS-330	Intelligence artificielle	Faltings	IN										2	2			4		sem P	
COM-308	Internet analytics	Grossglauser	SC										2	1	2		5		sem P	
CS-211	Introduction à l'informatique visuelle	Dillenbourg/Salzmann	IN	6			2		2								4		E	écrit
BIO-109	Introduction aux sciences du vivant (pour IC)	Zufferey R.	SV	7							4	2						6	E	écrit
CS-341	Introduction to computer graphics	Pauly	IN	6							2	1	2					6	E	écrit
COM-300	Modèles stochastiques pour les communications	Thiran P.	SC	4	8						4	2						6	H	écrit
CS-328	Numerical methods for visual computing	Jakob/Rhodin	IN	6							2	1						4	H	écrit
CS-323a	Operating systems implementation	Zwaenepoel	IN	1												2	4		sem P	
COM-302	Principles of digital communications	Telatar	SC	4							4	2						6	E	écrit
CS-309	Projet de systems-on-chip	Beuchat	IN	2												3	3		sem P	
COM-303	Signal processing for communications	Prandoni	SC	5	8						4	2						6	E	écrit
COM-309	Traitement quantique de l'information	Macris	SC								3	1						4	H	écrit
	Enseignement sciences humaines et sociales (SHS)																		8		
HUM-nnn	SHS : Cours à choix I selon Plan d'études SHS	Divers enseignants	SHS		2												2		sem A	
HUM-nnn	SHS : Cours à choix II selon Plan d'études SHS	Divers enseignants	SHS					2									2		sem P	
HUM-nnn	SHS : Cours à choix III selon Plan d'études SHS	Divers enseignants	SHS								2							2	sem A	
HUM-nnn	SHS : Cours à choix IV selon Plan d'études SHS	Divers enseignants	SHS										2					2	sem P	
	Totaux :																60	60		

Remarque :

* Se référer à l'art. 3 al. 4 du règlement d'application

Filières obligatoires :
- Software Construction (1)
- Computer Engineering (2)
- Computer Science Theory (3)

Filières optionnelles :
- Networks and Communications (4)
- Signal and Image Processing (5)
- Visual Computing (6)
- Biocomputing (7)
- Data Science (8)

2017-2018 **INFORMATIQUE**   **Cycle Bachelor**

Code	Matières	Enseignants sous réserve de modification	Sections	Crédits 2e	Crédits 3e	Période des cours
Obligatoire	**Filière 1 " Software Construction "**			**27 / 31**		
CS-210	Functional programming *	Kuncak / Odersky	IN	5		A
CS-322	Introduction to database systems *	Ailamaki/Koch	IN		4	P
CS-323	Introduction to operating systems *	Zwaenepoel	IN		3	P
CS-323a	*Operating systems implementation*	*Zwaenepoel*	*IN*		*4*	P
CS-206	Parallelism and concurrency *	Kuncak / Odersky	IN	4		P
CS-207	Programmation orientée système *	Chappelier	IN	3		P
CS-207a	Projet programmation système *	Chappelier/Figueiredo	IN	2		P
CS-305	Software engineering *	Candea	IN		6	A
Obligatoire	**Filière 2 " Computer Engineering "**			**11 / 14**		
CS-209	Architecture des Systems-on-Chip*	Ienne	IN	4		P
CS-307	Introduction to multiprocessor architecture*	Falsafi	IN		3	A
CS-208	Computer architecture *	Stojilovic	IN	4		A
CS-309	*Projet de Systems-on-Chip*	*Beuchat*	*IN*		*3*	P
Obligatoire	**Filière 3 " Computer Science Theory "**			**10**		
CS-250	Algorithms*	Kapralov	IN	6		A
CS-251	Theory of computation*	Vishnoi	IN	4		P
Optionnel	**Filière 4 " Networks and Communications "**			**16**		
EE-205	Circuits and systems II*	Gastpar	SC	4		P
COM-300	Modèles stochastiques pour les communications*	Thiran P.	SC		6	A
COM-302	Principles of digital communications*	Telatar	SC		6	P
Optionnel	**Filière 5 " Signal and Image Processing "**			**13 / 17**		
MATH-207b	*Analyse IV (cours recommandé, mais non obligatoire pour IN)*	*Nguyen H.-M.*	*MA*	4		P
EE-204	Circuits et systèmes I*	Rachidi	EL	3		A
EE-205	Circuits and systems II*	Gastpar	SC	4		P
CM-303	Signal processing for communications*	Prandoni	SC		6	P
Optionnel	**Filière 6 " Visual Computing "**			**14**		
CS-211	Introduction à l'informatique visuelle*	Dillenbourg/Salzmann	IN	4		P
CS-341	Introduction to computer graphics*	Pauly	IN		6	P
CS-328	Numerical methods for visual computing *	Jakob/Rhodin	IN		4	A
Optionnel	**Filière 7 " Biocomputing "**			**12**		
CH-nnn	Chimie générale avancée* (cours selon disponibilité horaire)	Divers ensignants	CGC		6	A
BIO-107	Introduction aux sciences du vivant (pour IC) *	Zufferey R.	SV		6	P
Optionnel	**Filière 8 " Data Science "**			**16 / 20**		
MATH-207b	*Analyse IV (cours recommandé, mais non obligatoire pour IN)*	*Nguyen H.-M.*	*MA*	4		P
EE-205	Circuits and systems II*	Gastpar	SC	4		P
COM-300	Modèles stochastiques pour les communications*	Thiran P.	SC		6	A
COM-303	Signal processing for communications*	Prandoni	SC		6	P

* cours obligatoire dans la filière

en italique : cours optionnel dans la filière

2017-2018 **INFORMATIQUE - Obligatoire**   **Cycle Master**

Code	Matières	Enseignants sous réserve de modification	Sections	Spécialisations	Semestres MA1 c	e	p	MA2 c	e	p	Crédits	Période des épreuves *	Type examen *
	Groupe "Core courses et options"										42		
	Groupe 1 "Core courses"										min. 15		
CS-450	Advanced algorithms	Svensson	IN	B C D E I				4	2	1	7	E	écrit
CS-470	Advanced computer architecture	Ienne	IN	A D G				2		2	4	E	oral
COM-401	Cryptography and security	Vaudenay	SC	D E J	4	2					7	H	écrit
CS-422	Database systems	Ailamaki	IN	B C G J				3	2	2	7	sem P	
CS-451	Distributed algorithms	Guerraoui	SC	C G I J	2	1					4	H	écrit
CS-423	Distributed information systems	Aberer	SC	B E J				2	1		4	E	écrit
CS-452	Foundations of software	Odersky	IN	C G	2	2					4	H	écrit
COM-404	Information theory and coding	Telatar	SC	B H I	4	2					7	H	écrit
CS-433	Machine learning	Jaggi/Urbanke	IN/SC	B F I J	4	2					7	H	écrit
	Groupe 2 "Options"	**(la somme des crédits des groupes 1 et 2 doit être de 42 crédits au minimum)**											
	Cours à option	Divers enseignants	Divers										
	Bloc "Projet et SHS" :										18		
CS-498	Projet en Informatique II		IN		←			2		→	12	sem A ou P	
HUM-nnn	SHS : introduction au projet	Divers enseignants	SHS		2		1				3	sem A	
HUM-nnn	SHS : projet	Divers enseignants	SHS							3	3	sem P	sans retrait
	Total des crédits du cycle master :										60		

Remarque :

* Se référer à l'art. 3 al. 4 du règlement d'application

Stage d'ingénieur :

Voir les modalités dans le règlement d'application

Mineurs :

Le cursus peut être complété par un des mineurs figurant dans l'offre de l'EPFL (renseignements à la page sac.epfl.ch/mineurs),

à l'exclusion des mineurs "Computer engineering", "Informatique", "Information security" et "Systèmes de communication" qui ne peuvent pas être choisis.

Parmi les mineurs offerts par l'EPFL, la section recommande à ses étudiants les mineurs suivants :

- Biocomputing (SIN)
- Computational science and Engineering (SMA)
- Management de la technologie et entrepreunariat (SMTE)
- Technologies biomédicales (SMT)
- Technologies spatiales (SEL)

Le choix des cours de tous les mineurs se fait sur conseil de la section de l'étudiant et du responsable du mineur.

Spécialisations uniquement pour les masters en 120 crédits :

A Computer Engineering-SP   F Signals, Images, and Interfaces
B Data Analytics   G Software Systems
C Foundations of Software   H Wireless Communication
D Information Security-SP   I Computer Science Theory
E Networking and Mobility   J Internet Information Systems

2017-2018 INFORMATIQUE - Options Cycle Master

Code	Matières	Enseignants sous réserve de modification	Sections	Spécialisations	M1 c	M1 e	M1 p	M2 c	M2 e	M2 p	Crédits	Période des épreuves *	Type examen *	Cours biennaux donnés en
	Options													
CS-420	Advanced compiler construction	Schinz	IN	A C G				2		2	4	sem P		
CS-440	Advanced computer graphics	Jakob	IN	F				2	1		6	sem P		
CS-471	Advanced multiprocessor architecture	Falsafi	IN	A G	4						6	sem A		2017-2018
COM-417	Advanced probability and applications	Lévêque	SC	H I				3	2		6	E	écrit	
CS-437	Algebraic coding theory	Shokrollahi	IN	D				2	2		4	sem P		
MATH-409	Algebraic curves and cryptography (pas donné en 2017-2018)	Jetchev	MA	D	2	2					5	H	écrit	2018-2019
CS-435	Analytic algorithms	Vishnoi	IN	B I				2	1		4	sem P		2017-2018
MATH-474	Applied biostatistics	Goldstein	MA					2	2		5	E	écrit	
CS-401	Applied data analysis	West	IN	B	2		2				6	H	écrit	
CS-456	Artificial neural networks	Gerstner	IN					2	1		4	E	écrit	
COM-415	Audio signal processing and virtual acoustics	Faller/Kolundzija	SC	F	2	2					4	H	écrit	
EE-554	Automatic speech processing	Bourlard	EL	F	2	1					3	H	écrit	
BIO-465	Biological modeling of neural networks	Gerstner	IN					2	2		4	E	écrit	
EE-512	Biomedical signal processing	Vesin	EL	F	4		2				6	H	écrit	
CS-490	Business design for IT services	Wegmann	SC					3			3	E	oral	
BIO-105	Cellular biology and biochemistry for engineers	Hirling	SV		2	2					4	H	écrit	
CS-413	Computational photography	Süsstrunk	SC	F				2		2	5	sem P		
CS-442	Computer vision	Fua	IN	F				2	1		4	E	écrit	
CS-453	Concurrent algorithms	Guerraoui	SC	C E G I	2	1					4	H	écrit	
CS-454	Convex optimization and applications	Lebret	MTE	H				1	2		4	sem P		
COM-480	Data visualization	Benzi	SC	B	2		2				4	sem A		
CS-438	Decentralized systems engineering	Ford	IN		2	2	2				6	sem A		
CS-472	Design technologies for integrated systems	De Micheli	IN	A	3		2				6	sem A		
CS-446	Digital 3D geometry processing	Pauly	IN	F	2	1	1				5	H	écrit	
CS-411	Digital education & learning analytics	Dillenbourg/Jermann	IN		2		2				4	H	oral	
ENG-466	Distributed intelligent systems	Martinoli	SIE	A	2	3					5	H	écrit	
COM-502	Dynamical system theory for engineers	Thiran P.	SC		2	1					4	H	écrit	
CS-473	Embedded systems	Beuchat	IN	A	2		2				4	H	oral	
CS-491	Enterprise and service-oriented architecture	Wegmann	SC	J				6			6	E	oral	
MATH-483	Gödel and recursivity (pas donné en 2017-2018)	Duparc	MA	I	2	2					5	H	écrit	2018-2019
CS-486	Human-computer interaction	Pu	IN	J				2	1	1	4	sem P		
EE-550	Image and video processing	Ebrahimi	EL	F	4		2				6	H	oral	
MICRO-511	Image processing I	Unser/Van De Ville	MT	F	3						3	H	écrit	
MICRO-512	Image processing II	Unser/Van De Ville	MT	F				3			3	E	écrit	
CS-487	Industrial automation	Pignolet/Tournier	SC					2		1	3	E	oral	
COM-402	Information security and privacy	Ford	IN	B D E				2	2		6	E	écrit	
CS-430	Intelligent agents	Faltings	IN	J	3	3					6	sem A		
CS-431	Introduction to natural language processing	Rajman/Chappelier	IN	B J				2	2		4	E	écrit	
COM-516	Markov chains and algorithmic applications	Lévêque/Macris	SC	B I	2	2					4	E	écrit	
COM-514	Mathematical foundations of signal processing	Kolundzija/Scholefield/Parhizkar	SC	F	3	2					6	H	écrit	
COM-405	Mobile networks	Hubaux	SC	D E G H				2	1		4	E	écrit	
COM-430	Modern digital communications: a hands-on approach	Rimoldi	SC	E F H	2		2				6	sem A		
COM-512	Networks out of control	Thiran P./ Celis	SC	B E H J				2	1		4	E	écrit	2017-2018
MATH-489	Number theory in cryptography	Jetchev	MA	D				2	2		5	E	écrit	2017-2018
CS-439	Optimization for machine learning	Jaggi	IN					2	2		4	E	écrit	
CS-596	Optional project in computer science	Divers enseignants	IN				← 2 →				8	sem A ou P		
COM-503	Performance evaluation (pas donné en 2017-2018)	Le Boudec	SC	B C E G J				3	1	2	7	E	oral	2018-2019
CS-489	Personal interaction studio	Huang	IN	F				2		4	6	sem P		
CS-522	Principles of computer systems	Argyraki/Candea	SC/IN	A C G	4						7	sem A		
MATH-467	Probabilistic method (pas donné en 2017-2018)	Pach	MA	I				2	2		5	E	écrit	2018-19
CS-476	Real-time embedded systems	Beuchat	IN	A				2		2	4	sem P		
COM-413	Real-time networks	Decotignie	SC	E				2			3	E	oral	
EE-511	Sensors in medical instrumentation	Aminian	EL	F				2	1		3	E	écrit	
MATH-318	Set theory	Duparc	MA		2	2					5	H	écrit	2017-2018
EE-472	Smart grid technologies	Paolone	EL					2	1	2	5	E	écrit	
EE-593	Social media	Gillet	EL	J				1		1	2	sem P		
COM-421	Statistical neurosciences (pas donné en 2017-2018)	Gastpar	SC	B				2	2		4	E	écrit	
COM-500	Statistical signal and data processing through applications	Ridolfi	SC	B F H				3	2		6	E	écrit	
CS-448	Sublinear algorithms for big data analysis	Kapralov	IN					3			4	sem P		
CS-550	Synthesis, analysis and verification (pas donné en 2017-2018)	Kuncak	IN	A C				2	2	2	6	sem P		2018-2019
COM-407	TCP/IP networking	Le Boudec	SC	D E G H	2	2					5	H	écrit	
CS-410	Technology ventures in IC (pas donné en 2017-2018)	Bugnion	IN					2		2	4	sem P		
CS-455	Topics in theoretical computer science (pas donné en 2017-2018)	Svensson	IN	B I	3	1					4	sem A		
CS-444	Virtual reality	Boulic	IN	F				2	1		4	sem P		

Remarque :
* Se référer à l'art. 3 al. 4 du règlement d'application

2017-2018   # INFORMATIQUE - spécialisations

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.

Code		Matières	Enseignants	Sections	Crédits	Période des cours	
		Spécialisation A "COMPUTER ENGINEERING-SP"	**Responsable : Prof. P. Ienne**		**62**		
CS-420		Advanced compiler construction	Schinz	IN	4		P
CS-470		Advanced computer architecture	Ienne	IN	4		P
CS-471		Advanced multiprocessor architecture	Falsafi	IN	6	A	
EE-431	*	Advanced VLSI design	Leblebici/Burg	EL	4		P
CS-472		Design technologies for integrated systems	De Micheli	IN	6	A	
ENG-466		Distributed intelligent systems	Martinoli	SIE	5	A	
CS-473		Embedded systems	Beuchat	IN	4	A	
EE-429	*	Fundamentals of VLSI Design	Leblebici/Burg	EL	4	A	
EE-432	*	Hardware systems modeling I	Vachoux	EL	2	A	
EE-433	*	Hardware systems modeling II	Vachoux	EL	2		P
EE-490b	*	Lab in EDA based design	Leblebici, Vachoux, Koukab	EL	4	A	
CS-522		Principles of computer systems	Argyraki/Candea	IN/SC	7	A	
CS-476		Real-time embedded systems	Beuchat	IN	4		P
CS-550		Synthesis, analysis and verification (pas donné en 2017-2018)	Kuncak	IN	6		P
		Spécialisation B "DATA ANALYTICS"	**Responsable : Prof. M. Grossglauser**		**86**		
CS-450		Advanced Algorithms	Svensson	IN	**7**		P
CS-435		Analytic Algorithms	Vishnoi	IN	4		P
CS-???		Applied Data Analysis	West	SC	6	A	
CS-422		Database Systems	Ailamaki	IN	7		P
COM-480		Data visualization	Benzi	SC	4	A	
CS-423		Distributed information systems	Aberer	SC	4		P
COM-402		Information Security and Privacy	Ford	IN	6		P
COM-404		Information theory and Coding	Telatar	SC	7	A	
COM-308	*	Internet analytics	Grossglauser	SC	5		P
CS-431		Introduction to natural language processing	Rajman/Chappelier	IN	4		P
COM-516		Markov chains and algorithmic applications	Lévêque/Macris	SC	4	A	
COM-512		Networks out of Control	Thiran P./ Celis	SC	4		P
CS-433		Machine learning	Jaggi/Urbanke	IN	7	A	
COM-503		Performance evaluation (pas donné en 2017-2018)	Le Boudec	SC	7		P
COM-421		Statistical neuroscience (pas donné en 2017-2018)	Gastpar	SC	4		P
COM-500		Statistical signal and data processing through applications	Ridolfi	SC	6		P
CS-455		Topics in theoretical computer science (pas donné en 2017-2018)	Svensson	IN	4	A	
		Spécialisation C "FOUNDATIONS OF SOFTWARE"	**Responsable : Prof. M. Odersky**		**56**		
CS-450		Advanced Algorithms	Svensson	IN	7		P
CS-420		Advanced compiler construction	Schinz	IN	4		P
CS-453		Concurrent algorithms	Guerraoui	SC	4	A	
CS-422		Database Systems	Ailamaki	IN	7		P
CS-451		Distributed algorithms	Guerraoui	SC	4	A	
UNIL-403	*	Emerging distributed architectures	Garbinato	HEC	6		P
CS-452		Foundations of software	Odersky	IN	4	A	
COM-503		Performance evaluation (pas donné en 2017-2018)	Le Boudec	SC	7		P
CS-522		Principles of computer systems	Argyraki/Candea	SC/IN	7	A	
CS-550		Synthesis, analysis and verification (pas donné en 2017-2018)	Kuncak	IN	6		P
		Spécialisation D "INFORMATION SECURITY-SP"	**Responsable : Prof. A. Lenstra**		**71**		
CS-450		Advanced algorithms	Svensson	IN	7		P
CS-470		Advanced computer architecture	Ienne	IN	4		P
COM-501	*	Advanced cryptography	Vaudenay	SC	4		P
EE-431	*	Advanced VLSI design	Leblebici/Burg	EL	4		P
CS-437		Algebraic Coding theory	Shokrollahi	IN	4		P
MATH-409		Algebraic curves and cryptography (pas donné en 2017-2018)	Jetchev	MA	5	A	
UNIL-402	*	Biometrics	Drygajlo	UNIL	3	A	
COM-401		Cryptography and security	Vaudenay	SC	7	A	
EE-429	*	Fundamentals of VLSI Design	Leblebici/Burg	EL	4	A	
COM-402		Information security and privacy	Ford	IN	6		P
EE-552	*	Media security	Ebrahimi	EL	6		P
COM-405		Mobile networks	Hubaux	SC	4		P
MATH-489		Number theory in cryptography	Jetchev	MA	5		P
COM-506	*	Student seminar : security protocols and applications	Oechslin/Vaudenay	SC	3		P
COM-407		TCP/IP Networking	Le Boudec	SC	5	A	

* = cours hors plan d'études pour les étudiants ne faisant pas la spécialisation

2017-2018 # INFORMATIQUE - spécialisations

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.

Code		Matières	Enseignants	Sections	Crédits	Période des cours
		Spécialisation E "NETWORKING AND MOBILITY"	**Responsable : Prof. J.-Y. Le Boudec**		**60**	
CS-450		Advanced algorithms	Svensson	IN	7	P
CS-453		Concurrent algorithms	Guerraoui	SC	4	A
COM-401		Cryptography and security	Vaudenay	SC	7	A
CS-423		Distributed information systems	Aberer	SC	4	P
COM-402		Information security and privacy	Ford	IN	6	P
COM-405		Mobile networks	Hubaux	SC	4	P
COM-430		Modern digital communications : A hands-on approach	Rimoldi	SC	6	A
COM-512		Networks out of control	Thiran P./Celis	SC	4	P
COM-503		Performance evaluation (pas donné en 2017-18)	Le Boudec	SC	7	P
COM-413		Real-time networks	Decotignie	SC	3	P
COM-414	*	Satellite communications systems and networks	Farserotu	SC	3	A
COM-407		TCP/IP networking	Le Boudec	SC	5	A
		Spécialisation F "SIGNAL, IMAGES AND INTERFACES"	**Responsable : Prof. M. Vetterli**		**95**	
CS-440		Advanced computer graphics	Jakob	IN	6	P
COM-415		Audio signal processing and virtual acoustics	Faller/Kolundzija	SC	4	A
EE-554		Automatic speech processing	Bourlard	EL	3	A
EE-512		Biomedical signal procesing	Vesin	EL	6	A
CS-413		Computational photography	Süsstrunk	SC	5	P
CS-442		Computer vision	Fua	IN	4	P
CS-446		Digital 3D Geometry Processing	Pauly	IN	5	A
EE-550		Image and video processing	Ebrahimi	EL	6	A
MICRO-511		Image processing I	Unser/Van De Ville	MT	3	A
MICRO-512		Image processing II	Unser/Van De Ville	MT	3	P
CS-341	*	Introduction to computer graphics	Pauly	IN	6	P
CS-433		Machine learning	Jaggi/Urbanke	IN/SC	7	A
COM-514		Mathematical foundations of signal processing	Kolundzija/Scholefield/Parhizkar	SC	6	A
COM-430		Modern digital communications : A hands-on approach	Rimoldi	SC	6	A
CS-489		Personal interaction studio	Huang	IN	6	P
EE-511		Sensors in medical instrumentation	Aminian	EL	3	P
COM-303	*	Signal processing for communications	Prandoni	SC	6	P
COM-500		Statistical signal and data processing through applications	Ridolfi	SC	6	P
CS-444		Virtual reality	Boulic	IN	4	P
		Spécialisation G "SOFTWARE SYSTEMS"	**Responsable : Prof. G. Candea**		**56**	
CS-420		Advanced compiler construction	Schinz	IN	4	P
CS-470		Advanced computer architecture	Ienne	IN	4	P
CS-471		Advanced multiprocessor architecture	Falsafi	IN	6	A
CS-453		Concurrent algorithms	Guerraoui	SC	4	A
CS-422		Database Systems	Ailamaki	IN	7	P
CS-451		Distributed algorithms	Guerraoui	SC	4	A
CS-452		Foundations of software	Odersky	IN	4	A
COM-405		Mobile networks	Hubaux	SC	4	P
COM-503		Performance evaluation (pas donné en 2017-2018)	Le Boudec	SC	7	P
CS-522		Principles of computer systems	Argyraki/Candea	IN	7	A
COM-407		TCP/IP networking	Le Boudec	SC	5	A
		Spécialisation H "WIRELESS COMMUNICATIONS"	**Responsable : Prof. E. Telatar**		**58**	
COM-510	*	Advanced digital communications (pas donné en 2017-2018)	Urbanke/Hassani	SC	7	A
COM-417		Advanced probability and applications	Lévêque	SC	6	P
CS-454		Convex optimization and applications	Lebret	MTE	4	P
COM-404		Information theory and coding	Telatar	SC	7	A
EE-445	*	Microwaves	Skrivervik	EL	3	A
COM-405		Mobile networks	Hubaux	SC	4	P
COM-430		Modern digital communications : A hands-on approach	Rimoldi	SC	6	A
COM-512		Networks out of control	Thiran P./Celis	SC	4	P
EE-345	*	Rayonnement et antennes	Skrivervik	EL	3	A
COM-414	*	Satellite communications systems and networks	Farserotu	SC	3	A
COM-500		Statistical signal and data processing through applications	Ridolfi	SC	6	P
COM-407		TCP/IP networking	Le Boudec	SC	5	A

Légende :

* = cours hors plan d'études pour les étudiants ne faisant pas la spécialisatio A = automne, P = printemps - 1 semestre comprend 14 semaines

2017-2018   **INFORMATIQUE - spécialisations**

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.

Code		Matières	Enseignants	Sections	Crédits	Période des cours	
		Spécialisation I. "COMPUTER SCIENCE THEORY"	**Responsable : Prof. N. Vishnoi**		**82**		
CS-450		Advanced algorithms	Svensson	IN	7		P
COM-417		Advanced probability and applications	Lévêque	SC	6		P
CS-435		Analytic algorithms	Vishnoi	IN	4		P
MAT-460	*	Combinatorial optimization	Eisenbrand	MA	5	A	
MATH-472	*	Computational finance	Nobile/Kressner/Pulido	MA	5	A	
CS-453		Concurrent algorithms	Guerraoui	SC	4	A	
MATH-461	*	Convexity	Eisenbrand	MA	5	A	
CS-451		Distributed algorithms	Guerraoui	SC	4	A	
MATH-483		Gödel and recursivity (pas donné en 2017-18)	Duparc	MA	5		P
COM-404		Information theory and coding	Telatar	SC	7	A	
CS-433		Machine learning	Jaggi/Urbanke	IN/SC	7	A	
COM-516		Markov chains and algorithmic applications	Lévêque/Macris	SC	4	A	
COM-300	*	Modèles stochastiques pour les communications	Thiran	SC	6	A	
MATH-467		Probabilistic method (pas donné en 2017-2018)	Pach	MA	5		P
MATH-745	*	Randomized algorithms	Eisenbrand	MA	4	A	
CS-455		Topics in theoretical computer science (pas donné en 2017-2018)	Svensson	IN	4	A	
		Spécialisation J. "INTERNET INFORMATION SYSTEMS"	**Responsable : Prof. B. Faltings et Prof. K. Aberer**		**62**		
COM-401		Cryptography and security	Vaudenay	SC	7	A	
CS-422		Database systems	Ailamaki	IN	7		P
CS-451		Distributed algorithms	Guerraoui	SC	4	A	
CS-423		Distributed information systems	Aberer	SC	4		P
CS-491		Enterprise and service-oriented architecture	Wegmann	SC	6		P
CS-486		Human-computer interaction	Pu	IN	4		P
CS-430		Intelligent agents	Faltings	IN	6	A	
CS-431		Introduction to natural language processing	Rajman/Chappelier	IN	4		P
CS-433		Machine learning	Jaggi/Urbanke	IN/SC	7	A	
COM-512		Networks out of control	Thiran P./Celis	SC	4		P
COM-503		Performance evaluation (pas donné en 2017-2018)	Le Boudec	SC	7		P
EE-593		Social Media	Gillet	EL	2		P

Légende :

* = cours hors plan d'études pour les étudiants ne faisant pas la spécialisation

A = automne, P = printemps - 1 semestre comprend 14 semaines

Biocomputing
Mineur interdisciplinaire

Section d'Informatique
Responsable : Prof. M. Salathé

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.
Les cours déjà suivis au bachelor ou au master ne peuvent pas être pris également dans un mineur.

52 **crédits offerts**

Code	Matières	Enseignants	Livret des cours	Crédits	Période des cours
CS-456	Artificial neural networks	Gerstner	IN	4	P
BIO-465	Biological modeling of neural networks	Gerstner	IN	4	P
BIO-315	Biomolecular structure and mechanics	Dal Peraro	SV	4	P
CS-432	Computational motor control	Ijspeert	SV	4	P
CS-423	Distributed information systems	Aberer	SC	4	P
ENG-466	Distributed intelligent systems	Martinoli	SIE	5	A
COM-502	Dynamical system theory for engineers	Thiran P.	SC	4	A
MICRO-511	Image processing I	Unser/Van De Ville	MT	3	A
MICRO-512	Image processing II	Unser/Van De Ville	MT	3	P
BIO-109*	Introduction aux sciences du vivant (pour IC)	Zufferey R.	SC	6	P
CH-353	Introduction to electronic structure methods	Röthlisberger	CGC	4	A
CS-433	Machine learning	Jaggi/Urbanke	SC	7	A

* Ce cours (ou cours équivalent) est obligatoire pour ce mineur s'il n'a pas été pris au bachelor / *This course (or equivalent) is mandatory for this minor if not already taken into the bachelor.*

D'autres cours peuvent être pris en approbation avec le Prof. Marcel Salathé / *Other courses ca be taken, if approved ba Prof. Marcel Salathé*

Légende :

A = automne, P = printemps

1 semestre comprend 14 semaines.

Computer engineering
Mineur disciplinaire

Section d'Informatique
Responsable : prof. P. Ienne

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.
Les cours déjà suivis au bachelor ou au master ne peuvent pas être pris également dans un mineur.

49 **crédits offerts**

Codes	Matières (liste indicative)	Enseignants	Livret des cours	Crédits	Période des cours
CS-470	Advanced computer architecture	Ienne	IN	4	P
CS-471	Advanced multiprocessor architecture	Falsafi	IN	6	A
EE-431	Advanced VLSI design	Leblebici/Burg	EL	4	P
CS-209	Architecture des systems-on-chip (ou cours équivalent)	Ienne	IN	4	P
CS-208	Computer architecture (ou cours équivalent)	Stojilovic	IN	4	A
CS-472	Design technologies for intergrated systems	De Micheli	IN	6	A
ENG-466	Distributed intelligent systems	Martinoli	SIE	5	A
CS-473	Embedded systems	Beuchat	IN	4	A
EE-429	Fundamentals of VLSI design	Leblebici/Burg	EL	4	A
EE-432	Hardware systems modeling I	Vachoux	EL	2	A
EE-433	Hardware systems modeling II	Vachoux	EL	2	P
CS-476	Real-time embedded systems	Beuchat	IN	4	P
CS-550	Synthesis, analysis, and verification (pas donné en 2017-18)	Kuncak	IN	6	P

Légende :

A = automne, P = printemps

1 semestre comprend 14 semaines.

2017-2018 Informatique

Mineur disciplinaire

Section d'Informatique

Responsable : Mme Sylviane Dal Mas

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.
Les cours déjà suivis au bachelor ou au master ne peuvent pas être pris également dans un mineur.

134 crédits offerts

Code		Matières (liste indicative)	Prérequis	Enseignants	Livret des cours	Crédits	Période des cours	
CS-250	1	Algorithms		Kapralov	IN	6	A	
CS-208	2	Computer architecture		Stojilovic	IN	4	A	
COM-208	3	Computer networks		Argyraki	SC	5	A	
CS-207	4	Programmation orientée système		Chappelier	IN	3		P
CS-207a	4	Projet programmation système		Chappelier/Figueiredo	IN	2		P
CS-251	5	Theory of computation		Vishnoi	IN	4		P
CS-322	6	Introduction to database systems	4	Ailamaki/Koch	IN	4		P
COM-301	7	Sécurité des réseaux	3	Oechslin	SC	4	A	
CS-323	8	Introduction to operating systems	4	Zwaenepeoel	IN	3		P
CS-305	9	Software engineering	4	Candea	IN	6	A	
CS-209	10	Architecture des systems-on-chip	2	Ienne	IN	4		P
CS-321	11	Informatique du temps réel	4	Decotignie	SC	4	A	
CS-330	12	Intelligence artificielle	1	Faltings	IN	4		P
CS-341	13	Introduction to computer graphics		Pauly	IN	6		P
CS-450		Advanced algorithms		Svensson	IN	7		P
CS-470		Advanced computer architecture		Ienne	IN	4		p
COM-401		Cryptography and security	*	Vaudenay	SC	7	A	
CS-422		Database systems		Ailamaki	IN	7		P
CS-451		Distributed algorithms		Guerraoui	SC	4	A	
CS-423		Distributed information systems		Aberer	SC	4		P
CS-452		Foundations of software		Odersky	IN	4	A	
COM-404		Information theory and coding		Telatar	SC	7	A	
CS-433		Machine learning		Jaggi/Urbanke	IN	7	A	
CS-440		Advanced computer graphics		Jakob	IN	6		P
CS-442		Computer vision		Fua	IN	4		P
CS-430		Intelligent agents		Faltings	IN	6	A	
CS-596		Optional project in computer science	*	Divers enseignants	IN	8	A ou P	
CS-455		Topics in theoretical computer science	(pas en 17-18)	Svensson	IN	4	A	

* pour étudiants titulaires d'un Bachelor en Informatique ou en Systèmes de communication

Légende :

A = automne, P = printemps

1 semestre comprend 14 semaines.

RÈGLEMENT D'APPLICATION DU CONTRÔLE DES ÉTUDES DE LA SECTION D'INFORMATIQUE
pour l'année académique 2017-2018
du 22 mai 2017

La direction de l'École polytechnique fédérale de Lausanne

vu l'ordonnance sur la formation menant au bachelor et au master de l'EPFL du 14 juin 2004,
vu l'ordonnance sur le contrôle des études menant au bachelor et au master à l'EPFL du 30 juin 2015,
vu le plan d'études de la section d'Informatique

arrête:

Article premier - Champ d'application

Le présent règlement fixe les règles d'application du contrôle des études de bachelor et de master de la section d'Informatique qui se rapportent à l'année académique 2017-2018

Art. 2 – Étapes de formation

1   Le bachelor est composé de deux étapes successives de formation :
- le cycle propédeutique d'une année dont la réussite se traduit par 60 crédits ECTS acquis en une fois, condition pour entrer au cycle bachelor. Le cycle propédeutique est commun avec celui de la section de systèmes de communication.
- le cycle bachelor s'étendant sur deux ans dont la réussite implique l'acquisition de 120 crédits, condition pour entrer au master.

2   Le master est composé de deux étapes successives de formation :
- le cycle master d'une durée de 2 semestres dont la réussite implique l'acquisition de 60 crédits. Ce cycle peut être complété par un mineur ou une spécialisation, impliquant l'acquisition de 30 crédits supplémentaires.
- le projet de master, d'une durée de 17 semaines à l'EPFL ou de 25 semaines hors EPFL (industrie ou autre haute école) et dont la réussite se traduit par l'acquisition de 30 crédits. Il est placé sous la responsabilité d'un professeur ou MER affilié à la section d'informatique ou de systèmes de communication.

Art 3 – Sessions d'examen

1   Les branches de session sont examinées pendant les sessions d'hiver ou d'été. Elles sont mentionnées dans le plan d'études avec la mention H ou E.

2   Les branches de semestre sont examinées pendant le semestre d'automne ou le semestre de printemps. Elles sont mentionnées dans le plan d'études avec la mention sem A ou sem P.

3   Une branche annuelle, c'est-à-dire dont l'intitulé tient sur une seule ligne dans le plan d'étude, est examinée globalement pendant la session d'été (E).

4   Pour les branches de session, la forme écrite ou orale de l'examen indiquée pour la session peut être complétée par des contrôles de connaissances écrits ou oraux durant le semestre, selon indications de l'enseignant.

Chapitre 1 : Cycle propédeutique

Art. 4 - Examen propédeutique

1   L'examen propédeutique comprend des branches « Polytechniques » pour 32 coefficients et des branches « Spécifiques » pour 29 coefficients, distribuées indifféremment sur deux blocs.

2   Le premier bloc de branches correspond à 37 coefficients et le second bloc de branches correspond à 24 coefficients.

3   L'examen propédeutique est réussi lorsque :
- l'étudiant a obtenu, à l'issue de la session d'hiver, une moyenne égale ou supérieure à 3.50 dans le premier bloc, condition pour entrer au semestre de printemps, et
- qu'il a obtenu, à l'issue de la session d'été, une moyenne égale ou supérieure à 4.00 dans chacun des deux blocs, condition pour entrer au cycle bachelor.

4   L'étudiant qui échoue l'examen propédeutique ne sera pas autorisé l'année suivante à répéter les branches de semestre pour lesquelles il a obtenu une note égale ou supérieure à 4.00.

Chapitre 2 : Cycle bachelor

Art. 5 - Organisation

1   Les enseignements du bachelor sont répartis en cinq blocs, le groupe « projet », le groupe « options » et le bloc transversal SHS.

2   Le groupe « options » se compose de toutes les branches à option figurant dans la liste du plan d'études de 2ème année et 3ème année. 33 crédits doivent être obtenus individuellement dans le groupe « options », dont 5 crédits dans les options de 2ème année. Les crédits pris en supplément des 5 crédits exigés de 2ème année peuvent être validés comme crédits à options de 3ème année.

3   En 3ème année, des cours comptant pour un maximum de 10 crédits au total peuvent être choisis en dehors de la liste du plan d'études. Les cours pris en dehors de cette liste doivent être acceptés préalablement par le directeur de la section.

Art. 6 – Filières

1   Les filières obligatoires sont acquises automatiquement par la réussite des blocs A, B C, D et E.

2   Les étudiants doivent réussir au minimum une filière optionnelle parmi les cinq proposées.

3   Une filière est réussie lorsque tous les crédits des cours obligatoires de la filière sont obtenus individuellement.

Art. 7 - Examen de 2ème année

1   Les **14 crédits** du plan d'études sont obtenus lorsque le bloc A est réussi.

2   Les **17 crédits** du plan d'études sont obtenus lorsque le bloc B est réussi.

3   Les **20 crédits** du plan d'études sont obtenus lorsque le bloc C est réussi.

4   Les **5 crédits de 2ème année** du groupe « options » s'acquièrent de façon indépendante, par réussite individuelle de chaque branche.

Art. 8 - Examen de 3e année

1   Les **13 crédits** du plan d'études sont obtenus lorsque le bloc D est réussi.

2   Les **7 crédits** du plan d'études sont obtenus lorsque le bloc E est réussi.

3   Les **8 crédits** du groupe « projet » s'acquièrent de façon indépendante, par réussite individuelle du projet.

4   Les **28 crédits de 3ème année** du groupe « options » s'acquièrent de façon indépendante, par réussite individuelle de chaque branche.

Art. 9 - Examen de 2ème et 3ème années

Le bloc « SHS transversal » est réussi lorsque les **8 crédits** du plan d'études sont obtenus.

Chapitre 3 : Cycle master

Art. 10 - Organisation

1   Les enseignements du cycle master sont répartis en un bloc " Projets + SHS" et deux groupes dont les crédits doivent être obtenus de façon indépendantes. Ils peuvent donner lieu à l'obtention d'une spécialisation ou d'un mineur.

2   Le Bloc " Projets + SHS" est composé d'un projet de 12 crédits et de l'enseignement SHS.

3   Le groupe 1 « Core courses » est composé des cours de la liste du plan d'études dans la rubrique « Master ».

4   Le groupe 2 « Options » est composé
- des cours de la liste du groupe 2 « options » du plan d'études dans la rubrique « Master » ;
- des crédits surnuméraires obtenus dans le groupe 1 « Core courses » ;
- d'un projet optionnel de 8 crédits suivant l'alinéa 5 ;
- de cours hors plan d'études suivant l'alinéa 6 ;
- de cours liés à une spécialisation ou un mineur suivant l'art.13,

5   Le projet du bloc " Projets et SHS" et le projet optionnel du groupe 2 « Options » ne peuvent être effectués dans le même semestre.

6   Des cours, comptant pour un maximum de 15 crédits au total, peuvent être choisis en dehors de la liste des cours sur le plan d'études dans la rubrique « Master ». Le choix de ces cours doit être accepté préalablement par le directeur de la section qui peut augmenter le maximum de 15 crédits si la demande est justifiée.

Art. 11 - Examen du cycle master

1   Le bloc " Projets et SHS" est réussi lorsque **18 crédits** sont obtenus.

2   Le groupe « Core courses et Options », composé du groupe 1 « Core courses » et du groupe 2 « Options » est réussi lorsque **42 crédits** sont obtenus.

3   Le groupe 1 « Core courses » est réussi lorsqu'**au moins 15 crédits** sont obtenus et si l'une des branches à **7 crédits** est réussie.

4   L'acquisition de 30 crédits supplémentaires dans le groupe « Core courses et Options » permet d'obtenir une spécialisation ou un mineur.

Art. 12 - Enseignement SHS

Les deux branches SHS donnent chacune lieu à 3 crédits. L'enseignement du semestre d'automne introduit à la réalisation du projet du semestre de printemps. Pour autant qu'il considère que le motif est justifié, le Collège des Humanités peut déroger à cette organisation. Il peut également autoriser à ce qu'un étudiant réalise son projet sur un semestre qui ne suit pas immédiatement celui dans lequel a lieu l'enseignement d'introduction.

Art. 13 - Mineurs et spécialisations

1   Afin d'approfondir un aspect particulier de sa formation ou de développer des interfaces avec d'autres sections, l'étudiant peut choisir la formation offerte dans le cadre d'un mineur figurant dans l'offre de l'EPFL ou d'une spécialisation de la section d'Informatique.

2   Le choix des cours qui composent un mineur se fait avec la section d'informatique et avec le responsable du mineur. Les mineurs « Computer engineering », « Information security », « Informatique » et « Systèmes de communication » ne peuvent pas être choisis.

3   Le choix des cours qui composent une spécialisation est soumis, pour concertation à la section d'informatique.

4   L'étudiant annonce le choix d'un mineur à sa section au plus tard à la fin du premier semestre des études de master.

5   L'étudiant qui choisit une spécialisation dans la liste figurant dans le plan d'études s'inscrit au plus tard au début du deuxième semestre des études de master.

6   Un mineur ou une spécialisation est réussi quand 30 crédits au minimum sont obtenus parmi les branches avalisées.

Chapitre 4 : Stage et projet de master

Art. 14 – Stage d'ingénieur

1   Les étudiants commençant leur cycle master doivent effectuer un stage d'ingénieur durant leur master :
- soit un stage d'été de minimum 8 semaines
- soit un stage de 6 mois en entreprise (en congé durant un semestre)
- soit un Projet de Master de 25 semaines en entreprise (valide le stage et le Projet de Master)

2   Le stage peut être effectué dès le 2ème semestre du cycle master, mais avant le projet de master.

3   Le responsable du stage de la section évalue le stage, par l'appréciation « réussi » ou « non réussi ». Sa réussite sera une condition pour l'admission au projet de master. En cas de non réussite, il pourra être répété une fois, en règle générale dans une autre entreprise.

4   Il est validé avec les 30 crédits du projet de master.

5   Les modalités d'organisation et les critères de validation du stage font l'objet d'une directive interne à la section.

Chapitre 5 : Mobilité

Art. 15 – Périodes de mobilité autorisées

Les étudiants de la section d'informatique peuvent effectuer un séjour de mobilité en 3ème année de bachelor et/ou dans le cadre du projet de master.

Art. 16 - Conditions

1   Pour une mobilité en 3ème année de bachelor, l'étudiant doit avoir réussi l'examen propédeutique avec une moyenne minimale de 4,5 et ne pas avoir de retard dans l'acquisition des 60 crédits de la 2ème année de bachelor.

2   Pour une mobilité au projet de master, l'étudiant peut être admis conditionnellement s'il n'a pas plus de 8 crédits manquants au cycle master.

3   Des conditions spécifiques existant en fonction des destinations, l'accord du délégué à la mobilité est nécessaire pour partir en séjour de mobilité.

Au nom de la direction de l'EPFL

Le président, M. Vetterli
Le vice-président pour l'éducation, P. Vandergheynst

Lausanne, le 22 mai 2017

2017-2018 **INFORMATIQUE**   **Passerelle HES**

Code	Matière	Enseignants sous réserve de modification	Sections	Semestre AUT			Semestre PRI			Crédits	Période des épreuves	Type examen
				c	e	p	c	e	p			
	Proposition de cours 60 crédits du Bachelor pour passer au Master :											
	Branches de bases									33		
CS-250	Algorithms	Kapralov	IN	4	2					6	H	écrit
MATH-203c	Analyse III	Nguyen H-M	MA	2	2					4	H	écrit
CS-208	Computer architecture	Stojilovic	IN	2		2				4	sem A	
COM-208	Computer networks	Argyraki	SC	2	2					5	sem A	
PHYS-114	General physics II	Dil	PH	2	2					4	H	écrit
MATH-232	Probabilités et statistique	Thibaud	MA				4	2		6	E	écrit
CS-251	Theory of computation	Vishnoi	IN				2	2		4	sem P	
	Branches d'approfondissement									27		
CS-210	Functional programming	Kuncak / Odersky	IN	2	2					5	sem A	
CS-322	Introduction to database systems	Ailamaki /Koch	IN				2	1	1	4	E	écrit
CS-323	Introduction to operating systems	Zwaenepoel	IN				2	1		3	sem P	
CS-206	Parallelisme and concurrency	Kuncak / Odersky	IN				1	1	2	4	sem P	
CS-207	Programmation orientée système	Chappelier	IN				1	2		3	sem P	
CS-207a	Projet programmation système	Chappelier/Figueiredo	IN						2	2	sem P	
CS-305	Software engineering	Candea	IN	2		3				6	sem A	
	Totaux			16	10	5	10	9	7	60		
	Totaux par semaine				31			26				
	Totaux par semestre				434			364				

colonnes c/e/p : nb d'heures par semaine

1 semestre comprend 14 semaines.

type examination : voir règlement d'application

RÈGLEMENT D'APPLICATION DU CONTRÔLE DES ÉTUDES CONCERNANT LA PASSERELLE HES-EPFL
pour l'année académique 2017-18
du 22 mai 2017

La direction de l'École polytechnique fédérale de Lausanne,

Vu la convention entre la CRUS (Conférence des recteurs des universités suisses), la KFH (Conférence suisse des recteurs des hautes écoles spécialisées) et la COHEP (Conférence suisse des rectrices et recteurs des hautes écoles pédagogiques) du 5 novembre 2007 ;
vu l'ordonnance concernant l'admission à l'EPFL du 8 mai 1995 ;
vu l'ordonnance sur la formation menant au bachelor et au master de l'EPFL du 14 juin 2004 ;
vu l'ordonnance sur le contrôle des études menant au bachelor et au master à l'EPFL du 30 juin 2015 ;

arrête :

Art. 1 - Passerelle HES-EPFL

1. Le présent règlement fixe les règles spécifiques à l'admission à la formation de master de l'EPFL sur la base d'un titre de bachelor HES (passerelle HES-EPFL ; ci-après la passerelle) qui se rapportent à l'année académique 2016-2017. S'appliquent au surplus les règles d'études générales à l'EPFL, en particulier celles figurant dans son ordonnance sur le contrôle des études.

2. Le titre de bachelor HES permet l'admission à un master EPFL dans la même discipline que le titre obtenu, avec condition de réussite de la passerelle.

3. La passerelle complète la formation HES par l'obtention à l'EPFL de 57 à 60 crédits en sciences de base, en ingénierie ou en architecture, suivant le master suivi.

4. Les branches permettant l'obtention des crédits de la passerelle sont divisées en un bloc de branches de base et un bloc de branches d'approfondissement. Chacun des deux blocs comprend entre 25 et 35 crédits.

5. La liste des branches et leur répartition dans chaque bloc sont déterminées par le directeur de la section du master suivi, selon les acquis qui ressortent du dossier académique du candidat.

6. Aux branches permettant d'acquérir les crédits de la passerelle, peuvent s'ajouter les éventuelles branches prérequises pour les branches du master suivi, conformément au livret de cours correspondant.

7. La réussite de la passerelle permet l'admission définitive au master. Elle ne donne lieu à aucun titre.

Art. 2 - Inscription anticipée au master

Pour s'inscrire aux branches de master, au moins 30 crédits doivent être acquis dans les branches de la passerelle (branches prises individuellement).

Art. 3 - Conditions de réussite de la passerelle

1. La passerelle est réussie lorsque ses crédits sont obtenus dans un délai de deux ans. Ces crédits sont obtenus par une moyenne des branches égale ou supérieure à 4.0 pour chacun des deux blocs de la passerelle.

2. L'obtention de moins de 30 crédits dans les branches de la passerelle (branches prises individuellement) au terme des examens de la première année entraîne un échec définitif.

Art. 4 - Règles applicables en deuxième année

1. Celui qui doit obtenir des crédits manquants sur une deuxième année demeure soumis au règlement de passerelle qui se rapporte à sa première année (année d'admission à la passerelle).

2. La répétition d'une branche est exécutée conformément aux règles de la branche pour l'année de la répétition.

Art. 5 - Période des cours et épreuves

1. Les cours de la passerelle débutent à la rentrée du semestre d'automne. L'entrée en cours d'année est exclue.

2. Conformément aux règles de l'EPFL, les branches de session sont examinées aux sessions d'examens d'hiver ou d'été correspondantes avec possibilité pour l'enseignant de prévoir des épreuves de semestre, les branches de semestre sont examinées pendant le semestre correspondant et les branches annuelles sont examinées à la session d'examens d'été ponctuant l'année académique.

Au nom de la direction de l'EPFL

Le président, M. Vetterli
Le vice-président pour l'éducation, P. Vandergheynst

Lausanne, le 22 mai 2017

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SECTION D'INFORMATIQUE

Cycle

Propédeutique

(1ère année)

2017 / 2018

CS-101	**Advanced information, computation, communication I**

Lenstra Arjen

Cursus	Sem.	Type
Information security minor	H	Opt.
Informatique	BA1	Obl.
Systèmes de communication	BA1	Obl.

Language	English
Coefficient	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Remarque

This course focuses on the foundational, discrete mathematics core of advanced computation.

Summary

Discrete mathematics is a discipline with applications to almost all areas of study. It provides a set of indispensable tools to computer science in particular. This course reviews (familiar) topics as diverse as mathematical reasoning, combinatorics, discrete structures & algorithmic thinking.

Content

I. Mathematical reasoning: propositional logic, propositional functions, quantifiers, rules of inference; this includes very basic logic circuits.
II. Sets and counting: cardinalities, inclusion/exclusion principle, sequences and summations.
III. Algorithms and complexity: basic algorithms, computational complexity, big-O notation and variants, countability.
IV. Number representations such as binary and hexadecimal and (postponed to 2nd semester) basic number theory: modular arithmetic, integer division, prime numbers, hash functions, pseudorandom number generation; applications.
V. Induction and recursion: mathematical induction, recursive definitions and algorithms.
VI. Basic combinatorial analysis: permutations, binomial theorem, basic generating functions.
VII. Basic probability: events, independence, random variables, Bayes' theorem.
VIII. Structure of sets: relations, equivalence relations, power set.
IX. (time permitting) Elementary graph theory: graphs, Euler and Hamilton paths, Dijkstra's algorithm, spanning trees.

Keywords

Propositional logic, counting, complexity, big-O, number representations, sets, matrices, modular arithmetic, induction, basic probabilities, Bayes theorem, combinatorial analysis, recurrences, generating functions, countability, graph theory.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize if there is a mistake in a (simple) proof
- Apply general problem-solving techniques
- Recognize the mathematical structures present in applications
- Apply simple recursion and use it to design recursive algorithms
- Apply the tools studied in class to solve problems
- Demonstrate familiarity with mathematical reasoning
- Solve linear recurrences and use generating functions
- Argue about (un)countability
- Formulate complete, clear mathematical proofs

Transversal skills

- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking

Teaching methods

Ex cathedra (blackboard) lectures

Expected student activities

Studying the book, test your understanding by making the exercises, ask questions

Assessment methods

Midterm exam (30%) and final exam (70%), both mostly (and possibly exclusively) multiple choice

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	A list of students assistants and their contact data will be made available on the moodle page for this course, along with an assignment of each registered student to one of the student assistants.
	If you have a question, first contact the student assistant assigned to you. If that does not help, contact one of the teaching assistants (Marguerite Delcourt, Dusan Kostic and Benjamin Wesolowski). Furthermore, you are always welcome to stop by at my office (INJ330, no office hours, I'm available when I'm there) for any type of question related to this course or your study at EPFL.
	Never hesitate to ask questions before, during or after the lectures!

Resources

Bibliography

"Discrete Mathematics and Its Applications", Kenneth H. Rosen, 7th ed, McGraw-Hill 2012. (You should be able to find the pdf on the web.)

Ressources en bibliothèque

- Discrete mathematics and its applications / Rosen

Websites

- http://will be provided later, if any

Moodle Link

- http://will be provided later

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

COM-102   Advanced information, computation, communication II

Rimoldi Bixio

Cursus	Sem.	Type
Informatique	BA2	Obl.
Systèmes de communication	BA2	Obl.

Language	English
Coefficient	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

Text, sound, and images are examples of information sources stored in our computers and/or communicated over the Internet. How do we measure, compress, and protect the informatin they contain?

Content

I. How to measure information. Source and probability. Entropy per symbol. Source coding.

II. Cryptography and information security. Modular arithmetic, modern algebra and number theory. The Chinese remainder theorem and RSA.

III. Protecting information. A few finite fields. Linear speaces. Hamming distance. Linear codes. Reed-Solomon codes.

Keywords

Shannon's entropy
Linear codes
Reed-Solomon codes
Number theory
Asymmetric Cryptography, RSA

Learning Outcomes

By the end of the course, the student must be able to:
- Understand Shannon's entropy
- Construct an optimal code
- Understand elementary number theory
- Know what an abelian group is
- Recognize a hidden isomorphism
- Know how RSA works
- Know a few linear codes on simple finite fields

Transversal skills

- Take feedback (critique) and respond in an appropriate manner.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Ex cathedrra with exercises

Expected student activities

Homework (written and grades) ever week.

Assessment methods

Continuous evaluations 10% and final exam 90%

Resources

Bibliography

"Sciences de l'information", J.-Y. Le Boudec, R. Urbanke et P. Thiran, online

Ressources en bibliothèque

- Introduction aux sciences de l'information : entropie, compression, chiffrement et correction d'erreurs / Le Boudec

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=851

MATH-111(e)   **Algèbre linéaire**

Shokrollahi Mohammad Amin

Cursus	Sem.	Type
Informatique	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

L'objectif du cours est d'introduire les notions de base de l'algèbre linéaire et ses applications.

Contenu

1. Systèmes linéaires
2. Algèbre matricielle
3. Espaces vectoriels
4. Bases et dimension
5. Applications linéaires et matrices
6. Le déterminant d'une matrice
7. Valeurs propres, vecteurs propres, et diagonalisation
8. Produit scalaire
9. Matrices orthogonales et matrices symétriques

Mots-clés

espace vectoriel, linéarite, matrice, déterminant, orthogonalité, produit scalaire

Compétences requises

Cours prérequis indicatifs

cours de base

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Effectuer des calculs standards en algèbre linéaire et en interprêter les résultats;
- Définir des concepts théoriques relevants de l'algèbre linéaire et en donner des exemples illustratifs;
- Identifier des exemples de concepts théoriques relevants de l'algèbre linéaire;
- Construire rigoureusement un raisonnement logique simple;
- Identifier quelques liens entre l'algèbre linéaire et d'autres branches des mathématiques.

Méthode d'enseignement

Cours ex cathedra, exercices en salle

Méthode d'évaluation

examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

Algèbre linéaire et applications, David C. Lay, 4e edition, editeur: Pearson, ISBN: 978-2-7440-7583-4

Ressources en bibliothèque

• Algèbre linéaire et applications / Lay

Préparation pour

Algèbre Linéaire II; Analyse II

MATH-111(en)   **Algèbre linéaire (anglais)**

Pach János

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Sciences et technologies du vivant	BA1	Obl.
Systèmes de communication	BA1	Obl.

Language	English
Coefficient	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

The purpose of the course is to introduce the basic notions of linear algebra and its applications.

Content

1. Linear systems;
2. Matrix algebra;
3. Vector spaces;
4. Bases and dimension;
5. Linear applications and matrices;
6. Determinant of a matrix;
7. Eigenvalues and eigenvectors;
8. Inner product, orthogonality, quadratic forms.

Keywords

vector space, linearity, matrix, determinant, orthogonality, inner product

Learning Outcomes

By the end of the course, the student must be able to:

- Accurately make standard computations relevant to linear algebra and interpret the results;
- Define and provide illustrative examples of relevant theoretical notions;
- Identify examples of relevant theoretical notions;
- Construct a simple logical argument rigorously;
- Identify some connections between linear algebra and other branches of mathematics.

Teaching methods

Lectures and exercises in the classroom

Assessment methods

Written exam

Supervision

Office hours   No

| Assistants | Yes |
| Forum | No |

Prerequisite for

linear algebra II; Analysis II

MATH-111(pi)   **Algèbre linéaire (cours pilote)**

Deparis Simone

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Sciences et technologies du vivant	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Remarque

Cours pilote, merci de consulter http://go.epfl.ch/algebre-pilote avant de vous inscrire. En vous inscrivant, vous acceptez que des données sur votre apprentissage soient collectées.

Résumé

L'objectif du cours est d'introduire les notions de base de l'algèbre linéaire et ses applications. Cette classe pilote est donné à 30% sous forme inversée.

Contenu

1. Systèmes linéaires
2. Algèbre matricielle
3. Espaces vectoriels
4. Bases et dimension
5. Applications linéaires et matrices
6. Le déterminant d'une matrice
7. Valeurs propres, vecteurs propres, et diagonalisation
8. Produits scalaires et espaces euclidiens
9. Matrices orthogonales et matrices symétriques

Mots-clés

espace vectoriel, linéarite, matrice, déterminant, orthogonalité, produit scalaire

Compétences requises

Cours prérequis indicatifs
cours de base

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

• Effectuer des calculs standards en algèbre linéaire et en interpréter les résultats;
• Définir des concepts théoriques relevants de l'algèbre linéaire et en donner des exemples illustratifs;
• Identifier des exemples de concepts théoriques relevants de l'algèbre linéaire;
• Construire rigoureusement un raisonnement logique simple;

• Identifier quelques liens entre l'algèbre linéaire et d'autres branches des mathématiques.

Méthode d'enseignement

Cours ex cathedra, exercices en salle.
30% du cours est sous forme classe inversée. Pendant 5 semaines, l'étudiant devra se préparer aux séances en classe en suivant le cours sur le MOOC.

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

Algèbre linéaire et applications, David C. Lay, 5e edition, editeur: Pearson, ISBN 978-2-7613-9109-2 (pas besoin de MonLab)
Algèbre linéaire et applications, David C. Lay, 4e edition, editeur: Pearson, ISBN: 978-2-7440-7583-4

Ressources en bibliothèque

• Algèbre linéaire et applications / Lay

Sites web

• https://courseware.epfl.ch/courses/course-v1:EPFL+AlgebreLineaire+2017_T3/about

Liens Moodle

• http://moodle.epfl.ch/course/search.php?search=MATH-111%28pi%29
• http://moodle.epfl.ch/course/view.php?id=15414

Préparation pour

Suite des études en ingénierie.

MATH-101(e)   **Analyse I**

Lachowska Anna

Cursus	Sem.	Type
Informatique	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

Étudier les concepts fondamentaux d'analyse et le calcul différentiel et intégral des fonctions réelles d'une variable.

Contenu

- Raisonner, démontrer et argumenter en mathématiques
- Nombres, structures et fonctions
- Suites, limites et continuité
- Séries numériques
- Fonctions réelles et processus de limite
- Calcul différentiel et intégral

Mots-clés

nombres réels, fonction, suite numérique, suite convergente/divergente, limite d'une suite, sous-suite, fonction, limite d'une fonction, fonction continue, série numérique, série convergente/divergente, convergence absolue, dérivée, classe C^k, théorème(s) des accroissements finis, développement limité, série entière, intégrale de Riemann, primitive, théorème de la valeur moyenne

Acquis de formation

- Le but fondamental de ce cours est d'acquérir les compétences suivantes :
- Raisonner rigoureusement pour analyser des problèmes
- Choisir ou sélectionner les outils d'analyse pertinents pour résoudre des problèmes
- Identifier les concepts inhérents à chaque problème
- Appliquer efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- Se montrer capable d'analyser et de résoudre des problèmes nouveaux
- Résoudre les problèmes de convergence, de suites et de séries
- Maîtriser les techniques du calcul différentiel et intégral
- Parmi les outils de base, on trouve les notions de convergence, de suites et de séries. Les fonctions d'une variable seront étudiées rigoureusement, avec pour but une compréhension approfondie des techniques du calcul différentiel et intégral.

Méthode d'enseignement

Cours ex cathedra et exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non
Autres	Tutorat des exercices
	autres mesures à définir

Ressources

Bibliographie

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. Volume 1. PPUR, 2016.

Ressources en bibliothèque

- Calcul différentiel et intégral / Douchet & Zwahlen
- (version électronique)

MATH-101(de)   **Analyse I (allemand)**

Semmler Klaus-Dieter

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Sciences et technologies du vivant	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	allemand
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

Es werden die Grundlagen der Analysis sowie der Differential- und Integralrechnung von Funktionen einer reellen Veränderlichen erarbeitet.

Contenu

- Mathematisches Begründen, Beweisen, und Argumentieren
- Zahlen, Strukturen und Funktionen
- Folgen, Grenzwerte und Stetigkeit
- Reihen
- Reelle Funktionen und Grenzwerte
- Differential- und Integralrechnung

Mots-clés

Reelle Zahlen, Funktionen, Folge, konvergente/divergente Folge, Grenzwert einer Folge, Teilfolge, Grenzwert einer Funktion, stetige Funktion, Reihe, konvergente/divergente Reihe, absolute Konvergenz, Ableitung, Funktionsklasse C^k, Mittelwertsatz der Differentialrechnung, Taylor-Entwicklung, Potenzreihe, Riemann-Integral, Stammfunktion, Mittelwertsatz der Integralrechnung

Acquis de formation

- Le but fondamental de ce cours est d'acquérir les compétences suivantes :
- Raisonner rigoureusement pour analyser des problèmes
- Choisir ou sélectionner les outils d'analyse pertinents pour résoudre des problèmes
- Identifier les concepts inhérents à chaque problème
- Appliquer efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- Se montrer capable d'analyser et de résoudre des problèmes nouveaux
- Résoudre les problèmes de convergence, de suites et de séries
- Maîtriser les techniques du calcul différentiel et intégral
- Parmi les outils de base, on trouve les notions de convergence, de suites et de séries. Les fonctions d'une variable seront étudiées rigoureusement, avec pour but une compréhension approfondie des techniques du calcul différentiel et intégral.

Méthode d'enseignement

Vorlesungen und Tutorien

Méthode d'évaluation

Schriftliche Klausur

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non
Autres	Tutorat des exercices
	autres mesures à définir

Ressources

Bibliographie

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral.
Tronc commun: Analyse I, PPUR, 2016.

Polycopiés

disponible

Liens Moodle

• http:// wird mitgeteilt

Préparation pour

Analysis II

MATH-101(en)   **Analyse I (anglais)**
Patakfalvi Zsolt

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Sciences et technologies du vivant	BA1	Obl.
Systèmes de communication	BA1	Obl.

Language	English
Coefficient	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

We study the fundamental concepts of analysis, calculus and the integral of real-valued functions of a real variable.

Content

- Reasoning , proving and arguing in mathematics
- Numbers, structures and functions
- Sequences, limit and continuity
- Series of reals
- Real-valued functions of a real variable and convergence
- Differential Calculus and the Integral

Keywords

real numbers, function, sequence,convergent/divergent sequence, limit, subsequence, limit of a function, continuous function, series of real numbers, convergent/divergent series, absolute convergence, derivative, class C^k, mean value theorem, Taylor's theorem, Taylor series, Riemann integral, indefinite integral, intermediate valuetheorem

Learning Outcomes

- The intended learning outcomes of this course are that students acquire the following capacities:
- Reason rigorously to analyse problems
- Choose appropriate analytical tools for problem solving.
- Be able to conceptualise in view of the applications of analysis.
- Apply efficiently mathematical concepts for problem solving by means of examples and exercises
- Analyze and to solve new problems.
- Master the basic tools of analysis as, for example, notions of convergence, sequences and series.
- Studying rigorously real functions we intend that students will demonstrate a deep understanding of calculus

Teaching methods

Ex cathedra lecture and exercises in the classroom

Assessment methods

Written exam

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Tutoring of exercises
	other measures to be defined

MATH-106(e)   **Analyse II**

Lachowska Anna

Cursus	Sem.	Type
Informatique	BA2	Obl.
Systèmes de communication	BA2	Obl.

Langue	français
Coefficient	6
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

Étudier les concepts fondamentaux d'analyse et le calcul différentiel et intégral des fonctions réelles de plusieurs variables.

Contenu

- L'espace R^n
- Calcul différentiel des fonctions à plusieurs variables
- Intégrales multiples
- Équations différentielles ordinaires

Mots-clés

Espace vectoriel euclidien, , dérivée partielle, différentielle, matrice jacobienne, extremum local d'une fonction de plusieurs variables, matrice hessienne, développement limité, gradient, divergence, rotationnel, règle de composition, théorème des fonctions implicites, multiplicateurs de Lagrange, intégrale multiple, équation différentielle ordinaire

Compétences requises

Cours prérequis obligatoires

Analyse I, Algèbre linéaire I

Cours prérequis indicatifs

Analyse I, Algèbre linéaire I

Concepts importants à maîtriser

- calcul différentiel et intégral des fonctions réelles d'une variable
- les notions de convergence
- espace vectoriel, matrices, valeurs propres

Acquis de formation

- Le but fondamental de ce cours reste, comme pour la partie I, d'acquérir les capacités suivantes :
- Appliquer
- avec aisance et approfondir les compétences et connaissances acquises en Analyse I :
- Raisonner
- rigoureusement pour analyser les problèmes
- Choisir ou sélectionner
- les outils d'analyse pertinents pour résoudre des problèmes

- Identifier
- les concepts inhérents à chaque problème
- Appliquer
- efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- Se montrer capable d'analyser et de résoudre des problèmes nouveaux
- Maîtriser les techniques du calcul différentiel et intégral.
- Maîtriser les équations différentielles élémentaires, l'espace R^n, les fonctions de plusieurs variables, les dérivées partielles et les intégrales multiples.

Méthode d'enseignement

Cours ex cathedra et exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non
Autres	Tutorat des exercices autres mesures à définir

Ressources

Bibliographie

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. PPUR, 2011.
L'enseignant précisera les manuels recommandés dans son cours.

MATH-106(en)   **Analyse II (anglais)**

Mountford Thomas

Cursus	Sem.	Type
Chimie et génie chimique	BA2	Obl.
Génie civil	BA2	Obl.
Génie mécanique	BA2	Obl.
Génie électrique et électronique	BA2	Obl.
Informatique	BA2	Obl.
Microtechnique	BA2	Obl.
Science et génie des matériaux	BA2	Obl.
Sciences et ingénierie de l'environnement	BA2	Obl.
Sciences et technologies du vivant	BA2	Obl.
Systèmes de communication	BA2	Obl.

Language	English
Coefficient	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

The course studies fundamental concepts of analysis and the calculus of functions of several variables.

Content

-The Euclidean space R^n.
-Vector functions and curves
-Differentiation of functions of several variables.
-Multiple integrals
-Ordinary differential equations.

Keywords

Euclidean vector space, partial derivative,differential, Jacobian, Hessian, Taylor expansion, gradient, chain rule, implicit function theorem, Lagrange multipliers, multiple integrals, ordinary differential equation

Learning Prerequisites

Required courses

Analysis I, Linear Algebra I

Important concepts to start the course

-calculus of functions of one variable
-concepts of convergence
-vector space, matrices, eigenvalues

Learning Outcomes

• The goal of this course consists as for Analysis 1 is that students acquire the following capacities:

• Consolidate the skills and knowledge they acquired in Analysis 1.

• Reason

• rigorously and to analyse problems

• Choose

• appropriate analytical tools for problem solving.

• Conceptualize problems

- Apply
- efficiently mathematical concepts for problem solving by means of examples and exercises
- Analyze
- and to solve new problems.
- Master the basic tools of analysis
- Master the basic tools of elementary ordinary differential equations, the Euclidean space R^n and functions of several variables

Teaching methods

Ex cathedra lectures, exercises sessions in the classroom.

Assessment methods

Written exam

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Tutoring of exercises
	other measures to be defined

Resources

Bibliography

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. PPUR, 2011.

Websites

- http://mcss.epfl.ch/page-105207-en.html

CS-173

Conception de systèmes numériques

Kluter Ties Jan Henderikus

Cursus	Sem.	Type
Informatique	BA2	Obl.
Systèmes de communication	BA2	Obl.

Langue	français
Coefficient	6
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Projet	2 hebdo

Résumé

Les étudiants se familiarisent avec les composants numériques de base des systèmes de traitement de l'information, et apprennent à utiliser des méthodes modernes d'analyse et de synthèse des systèmes combinatoires et séquentiels, à l'aide notamment des langages tels que VHDL.

Contenu

1. Numérique vs analogique. Logique : principes et opérateurs
2. Algèbre booléenne. Synthèse combinatoire. Table de Karnaugh
3. Technologie
4. Dispositifs combinatoires
5. Elément de mémoire. Bascule bistable
6. Machines séquentielles : modes de représentation et d'analyse
7. Machines séquentielles : synthèse
8. Les circuits programmables
9. Le langage VHDL pour la synthèse de systèmes numériques

Mots-clés

Systèmes numériques, portes logiques, algèbre booléenne, systèmes combinatoires, systèmes séquentiels, systèmes de numération, VHDL

Compétences requises

Cours prérequis obligatoires

Rien

Cours prérequis indicatifs

Rien

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Utiliser les principaux dispositifs logiques et arithmétiques des systèmes de traitement de l'information.
- Expliquer les modes de représentation des systèmes combinatoires et séquentiels.
- Utiliser des méthodes modernes de synthèse et de simplification des systèmes combinatoires et séquentiels.
- Utiliser un langage tel que VHDL pour la synthèse d'éléments numériques complexes.
- Expliquer les modes de représentation des principaux types de données.
- Choisir ou sélectionner les composants et les méthodes de synthèse pour concevoir un système numérique

complexe.
- Développer des systèmes numériques complexes.

Méthode d'enseignement
Cours ex cathedra, exercices et projets de laboratoire

Travail attendu
L'étudiant doit assister au cours, préparer et résoudre les exercices, préparer et réaliser les projets de laboratoire

Méthode d'évaluation
Test intermédaire (50%)
Examen final (50%)

Ressources

Bibliographie
W. J. Dally and R. C. Harting, *Digital design : A systems approach*, Cambridge University Press, 2012
J. Wakerly, *Digital design (4th edition)*, Prentice Hall, 2005
P. J. Ashenden, *The student's guide to VHDL (2nd edition)*, Morgan Kaufmann, 2008
C. Maxfield, *Bebop to the boolean boogie: An unconventional guide to electronics (3rd edition)*, Newnes, 2008

Ressources en bibliothèque
- Digital design : A systems approach / Dally
- Digital design / Wakerly
- The student's guide to VHDL
- Bebop to the boolean boogie / Maxfield

Polycopiés
Copies des transparents disponibles sur moodle

Préparation pour
Architecture des ordinateurs (CS-208)

CS-107

Introduction à la programmation

Sam Jamila

Cursus	Sem.	Type
Informatique	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	français
Coefficient	5
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	150h
Semaines	14
Heures	**5 hebdo**
Cours	2 hebdo
Exercices	3 hebdo

Résumé

Ce cours aborde les concepts fondamentaux de la programmation et de la programmation orientée objet (langage JAVA). Il permet également de se familisarier avec un environnement informatique (station de travail sous UNIX)

Contenu

• Introduction à l'environnement UNIX (connection, multi-fenêtrage, édition de textes, email, ...), éléments de base du fonctionnement d'un système informatique et prise en main d'un environnement de programmation (éditeur, compilateur, ...).

• Initiation à la programmation (langage JAVA) : variables, expressions, structures de contrôle, modularisation, entrées-sorties

• Introduction à la programmation objet (langage JAVA) : objets, classes, méthodes, encapsulation, héritage, polymorphisme

• Pratique de concepts algorithmques fondamentaux (récursion, recherche, tri etc.).

Mots-clés

Java, programmation orientée-objet, Unix.

Compétences requises

Cours prérequis obligatoires
Aucun

Cours prérequis indicatifs
Aucun

Concepts importants à maîtriser
Aucun

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Concevoir des algorithmes résolvant des tâches simples
- Transcrire un algorithme en son programme équivalent en Java
- Modéliser en langage Java une situation simple du monde réelle
- Structurer un problème complexe en sous-problèmes
- Analyser un code pour en décrire le résultat ou le corriger
- Argumenter la validité de décision de conception de base dans un programme orienté-objet
- Tester l'adéquation du résultat d'un programme par rapport à la tâche visée
- Réaliser de façon autonome une application de petite taille au moyen du langage Java et en utilisant les concepts fondamentaux de la programmation orientée objet

Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.
- Utiliser une méthodologie de travail appropriée, organiser un/son travail.
- Accéder aux sources d'informations appropriées et les évaluer.

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateur et support en ligne MOOC

Travail attendu

participation au cours, résolutions d'exercices.

Méthode d'évaluation

1- Examen écrit individuel (50%)
2- Mini-projet 1 (20%)
3- Mini-projet 2 (30%)
Les mini-projets se font à deux.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)
Oui

Bibliographie
Notes de cours disponibles en ligne. Livre(s) de référence indiqué(s) en début de semestre

Liens Moodle

- http://moodle.epfl.ch/enrol/instances.php?id=14847

Préparation pour

Pratique de la programmation orientée-objet (CS-108)

PHYS-101(c)　**Physique générale I**

Fuerbringer Jean-Marie

Cursus	Sem.	Type
Informatique	BA1	Obl.

Langue	français
Coefficient	5
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	150h
Semaines	14
Heures	**5 hebdo**
Cours	3 hebdo
Exercices	2 hebdo

Résumé

Le but du cours de physique générale est de donner à l'étudiant les notions de base nécessaires à la compréhension des phénomènes physiques. L'objectif est atteint lorsque l'étudiant est capable de prévoir quantitativement les conséquences de ces phénomènes avec des outils théoriques appropriés.

Contenu

Introduction et Cinématique : référentiels, trajectoire, vitesse, accélération, coordonnées cartésiennes et cylindriques.
Dynamique du point matériel : quantité de mouvement, lois de Newton, forces fondamentales, empiriques et de liaison, mouvement oscillatoire, moment cinétique.
Travail, puissance, énergie : énergies cinétique, potentielle, mécanique, lois de conservation, mouvements gravitationnels, collisions.
Changement de référentiels : dynamique dans les référentiels non inertiels
Dynamique des systèmes : centre de masse, moment cinétique, énergie
Solide indéformable : moment cinétique, moment d'inertie, effets gyroscopiques

Compléments

L'enseignement peut contenir, mais pas exclusivement, les éléments suivants: mécanique analytique, coordonnées sphériques, relativité restreinte

Mots-clés

Physique générale, mécanique du point matériel, mécanique du solide, coordonnées, cinématique, relativité, énergie, travail

Compétences requises

Cours prérequis indicatifs

• Niveau mathématique de la maturité fédérale, voir par exemple "www.vsmp.ch/crm/cat.htm"

• "Savoir-Faire en Maths - bien commencer ses études scientifiques", Y. Biollay, A. Chaabouni, J. Stubbe, PPUR, 2010

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
• Elaborer un modèle physique d'un système mécanique
• Démontrer un savoir-faire dans la résolution de problèmes
• Structurer les modèles en termes d'équations différentielles
• Formuler et utiliser des hypothèses simplificatrices pour décrire une expérience

- Utiliser les modèles théoriques qui décrivent la Nature
- Estimer les ordres de grandeur
- Relier les notions de cours et les observations du monde quotidien

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Cours, exercices en salle et travail personnel

Méthode d'évaluation

Examen écrit à la session d'hiver

Encadrement

Autres   Tutorat

Ressources

Bibliographie

Livre de cours:

• Traité de Physique: La Mécanique. J.-Ph. Ansermet, Presses polytechniques universitaires romandes, Deuxième édition largement remaniée, Septembre 2013 (ISBN 978-2-88915-024-3), Troisième édition avec codes QR renvoyant au MOOC, Septembre 2015.

MOOC:
- Mécanique de Newton, J.-Ph. Ansermet, www.coursera.org/course/mecanique, 2013

Autres références:
- Physique Générale (vol.1) 2ème édition, Alonso & Finn, InterEditions, Paris, 1988
- Physics for scientists and engineers, 4ème édition, Giancoli. International Edition, Prentice Hall
- Conceptual Physics, 10th edition, Paul G. Hewitt, City college San Francisco, 2005

Ressources en bibliothèque

- Mooc-Mécanique de l'EPFL /
- Physics for scientists and engineers / Giancoli
- Conceptual Physics / Hewitt
- Physique Générale / Alonso
- La Mécanique / Ansermet

Sites web

- http://moodle.epfl.ch/course/view.php?id=2531

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=2531

Préparation pour

Physique générale II

PHYS-101(de) **Physique générale I (allemand)**

Gruetter Rolf

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Mathématiques	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Sciences et technologies du vivant	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	allemand
Coefficient	5
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	150h
Semaines	14
Heures	**5 hebdo**
Cours	3 hebdo
Exercices	2 hebdo

Résumé

Die Studenten erwerben die Grundkenntnisse zum Verständnis von physikalischen Phänomenen der Mechanik. Sie entwickeln die Fähigkeit, die Auswirkungen dieser Phänomene mit den entsprechenden theoretischen Werkzeugen quantitativ zu bestimmen.

Contenu

Enleitung und Kinematik: Abschätzen von Grössenordnungen, physikalische Denkweise, Bezugssysteme, Bahnkurve, Geschwindigkeit, Beschleunigung, kartesische und zylindrische Koordinatensysteme
Dynamik des Massenpunktes: Impuls, Newton's Axiome, grundlegende und empirische Kräfte(verhältnisse), Mechanische Schwingungen, Drehimpuls.
Arbeit, Leistung, Energie: Kinetische, potentielle und mechanische Energie, Erhaltungssätze, Gravitation. Kollisionen.
Aenderung der Bezugssysteme: beschleunigte Bezugssysteme
Mechanik von Systemen: Massenzentrum, (Dreh)impuls, Energie
Mechanik starrer Körper: Drehimpuls, Trägheitsimpuls, Hebelgesetz, gyroskopische Effekte
Ergänzungen: Der Stoff kann folgende nicht-inklusive Elemente beinhalten: sphärische Koordinatensystem, Einführung in die spezielle Relativitätstheorie

Mots-clés

Allgemeine Physik, Koordinaten, Kinematik, Energie, Arbeit, Mechanik des starren Körpers, Koordinaten, Relativität.

Compétences requises

Cours prérequis indicatifs

Ausgezeichnete Grundkenntnisse der Mathematik Niveau Schweizerische Maturitätsprüfung (zB. http://www.math.ch/kanon/)

Concepts importants à maîtriser

Vektoralgebra: Skalar- und Vektorprodukt, Zerlegen von Vektoren. Beziehungen des rechtwinkligen Dreiecks.
Lösung von linearen Gleichungsystemen mit 2 oder 3 Unbekannten.
Integration/Differentiation von Funktionen und Vektoren.
Umwandlung physikalischer Einheiten

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Erstellen eines physikalischen Modells eines mechanischen Systems.
- Know-how entwickeln, um ein Problem zu lösen
- Erkennen der korrekten Einheiten
- Beurteilen benutzter vereinfachender Annahmen
- Identifizieren des vorhergesehenen qualtitativen Verhaltens
- Abschätzen von Größenordnungen
- Zusammenhänge zwischen Vorlesung und Alltag erkennen
- Berücksichtigen der signifikanten Stellen
- Herleiten der Bewegungsgleichungen

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Kurs, Ubungen im Saal und persönliche Arbeit.

Travail attendu

Neben der Vorlesung und Ubungs-sessionen (Diese Zeitangaben können von der geleisteten Vorbereitung aufs Studium und der Qualität der Arbeitsorganisation abhangen):
Vor der Vorlesung, ca. 2 bis 3 Seiten im Vorlesungswerk lesen (ca. 15 min)
Nach der Vorlesung, eine ZUsammenfassung/Formelsammlung erstellen (ca. 30 min.), gefolt von mehreren Vorbereitungsübungen (ca. 60 min)
Ubungen vervollständigen (ca. 3 h), online quiz beantworten (ca. 15 min)

Méthode d'évaluation

schriftliche Prüfung

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Ressources

Bibliographie

- Physics for scientists and engineers, 4ème édition, Giancoli. International Edition, Pearson/Prentice Hall
- Mathematics for physics with calculus, Das, Pearson/Prentice Hall

- W. Demtröder, Experimentalphysik 1, Springer Verlag

Ressources en bibliothèque

- Experimentalphysik / Demtröder

Polycopiés

Eine Spezialausgabe von Giancoli/Das ist in der Buchhandlung La Fontaine erhältlich.

Sites web

- http://lifmet.epfl.ch

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=14481

Préparation pour

Physik II

PHYS-101(en)   **Physique générale I (anglais)**
Bréchet Sylvain

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Mathématiques	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Sciences et technologies du vivant	BA1	Obl.
Systèmes de communication	BA1	Obl.

Language	English
Coefficient	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly

Summary

Students will learn the principles of mechanics to enable a better understanding of physical phenomena, such as the kinematics and dyamics of point masses and solid bodies. Students will acquire the capacity to quantitatively analyze these effects with the appropriate theoretical tools.

Content

The course may contain, but not exclusively, the following elements :

Mechanics

Introduction and kinematics
Reference frames, trajectories, velocity, acceleration, Cartesian, spherical and cylindrical coordinates.

Dynamics of the point mass and solid body
Momentum, Newton's laws, fundamental forces, empirical forces and constraints. Oscillatory motion, Angular momentum.

Work, power, energy
Kinetic energy, potential energy, conservation laws, gravitational motion. Collisions.

Keywords

General physics, point masses, coordinates, kinematics, energy, work

Learning Prerequisites

Recommended courses
Math level required for "maturité fédérale", see on the left the hyperlinks and the book, indicative of the level of math appropriate for a good start at EPFL.

Learning Outcomes

By the end of the course, the student must be able to:
- Develop a know-how to solve a problem
- Structure models in terms of differentials equations

- Apply simplifying assumptions to describe an experience
- Estimate orders of magnitude
- Distinguish the theoretical models describing Natura
- Contextualise theoretical models in every day life
- Formulate a physical model

Transversal skills

- Use a work methodology appropriate to the task.

Teaching methods

Lectures + exercises

Assessment methods

The course concludes with a written exam

Resources

Bibliography

- Serway, Physics for Scientists and Engineers.
- Douglas Giancoli. Physics for Scientists and Engineers. 4th Edition.
- D. Halliday, R. Resnick, K. S. Krane. Physics, Volume 1.

Ressources en bibliothèque

- La Mécanique / Ansermet
- Conceptual Physics / Hewitt
- Physique Générale / Alonso
- Physics for scientists and engineers / Giancoli
- Mooc-Mécanique / Ansermet

Prerequisite for

General physics II

CS-108   **Pratique de la programmation orientée-objet**

Schinz Michel

Cursus	Sem.	Type
Informatique	BA2	Obl.
Systèmes de communication	BA2	Obl.

Langue	français
Coefficient	9
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	270h
Semaines	14
Heures	**10 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Projet	6 hebdo

Résumé

Les étudiants perfectionnent leurs connaissances en Java et les mettent en pratique en réalisant un projet de taille conséquente. Ils apprennent à utiliser et à mettre en œuvre les principaux types de collections (listes, ensembles, tables associatives), et examinent quelques patrons de conception.

Contenu

Approfondissement des connaissances du langage Java, en particulier de la généricité (polymorphisme paramétrique), des classes imbriquées et anonymes et des lambdas.
Introduction à différents aspects de la bibliothèque standard Java : collections, entrées-sorties, interfaces utilisateur graphiques, etc.
Etude des mises en œuvre des collections par chaînage, arbres binaires de recherche ou hachage.
Introduction aux patrons de conception (*design patterns*) et examen des plus importants (*Decorator*, *Composite*, *Builder*, etc.).
Examen de l'utilisation judicieuse de l'héritage et de l'immuabilité.
Réalisation d'un projet de programmation conséquent en Java.

Mots-clés

Java, programmation orientée-objets, collections, patrons de conception.

Compétences requises

Cours prérequis obligatoires

Introduction à la programmation.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Concevoir et écrire des programmes Java de taille moyenne.
- Utiliser à bon escient la totalité des concepts de Java.
- Utiliser et concevoir des classes et méthodes génériques en Java.
- Utiliser et mettre en oeuvre les principales sortes de collection (listes, ensembles, tables associatives).
- Utiliser judicieusement l'héritage et l'immuabilité dans les langages orienté-objets.
- Reconnaitre et savoir utiliser plusieurs patrons de conception.

Compétences transversales

• Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.

Méthode d'enseignement

Ex-cathedra.

Travail attendu

Participation au cours, réalisation des exercices, réalisation du projet.

Méthode d'évaluation

Durant le semestre : projet (60%), examen intermédiaire (15%) et examen final (25%).

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)
Oui

Ressources en bibliothèque
• Effective Java / Bloch
• Java Generics and Collections / Naftalin

Sites web
• http://cs108.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SECTION D'INFORMATIQUE

Cycle Bachelor

(2$^{\text{ème}}$ année et 3$^{\text{ème}}$ année)

2017 / 2018

MATH-310   **Algèbre**

Testerman Donna

Cursus	Sem.	Type
Chimie	BA5	Opt.
HES -SC	H	Obl.
Information security minor	H	Opt.
Informatique	BA5	Opt.
Systèmes de communication	BA5	Obl.

Langue	français
Crédits	3
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Le but de ce cours est d'introduire et d'étudier certaines notions fondamentales de l'algèbre qui sont particulièrement utiles dans des applications.

Contenu

1. Basic arithmetic
2. Congruences and congruence classes
3. Rings and fields
4. Groups
5. Polynomials
6. Finite fields

Compétences requises

Cours prérequis indicatifs

Obligatoire pour IN/SC: Analyse III, Physique générale I, Physique générale II et Probabilités et statistique

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Construire des connaissances sur les bases de l'algèbre

Méthode d'enseignement

Cours ex-cathedra
Exercices

Méthode d'évaluation

Examen écrit

Ressources

Bibliographie

L.N. Childs "A concrete introduction to higher Algebra", Springer
Serge Lang "Undegraduate Algebra", Springer

Ressources en bibliothèque

- A concrete introduction to higher Algebra / Childs

Polycopiés

Sur moodle

CS-250	**Algorithms**		
	Kapralov Mikhail		

Cursus	Sem.	Type
HES - IN	H	Obl.
HES -SC	H	Obl.
Information security minor	H	Opt.
Informatique	BA3	Obl.
Mathématiques	BA5	Opt.
Mineur en Informatique	H	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Systèmes de communication	BA3	Obl.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

The students learn the theory and practice of basic concepts and techniques in algorithms. The course covers mathematical induction, techniques for analyzing algorithms, elementary data structures, major algorithmic paradigms such as dynamic programming, sorting and searching, and graph algorithms.

Content

Mathematical Induction

• Mathematical background, Euler's formula for trees, Schwartz-Zippel lemma.

Analysis of Algorithms

• O-notation, time and space complexity, recurrence relations, probabilistic analysis.

Data structures

• Arrays, linked lists, trees, heaps, hashing, graphs.

Design of algorithms by induction

• Evaluating polynomials, divide-and-conquer algorithms, dynamic programming.

Greedy Algorithms

• Spanning tree and shortest path algortihms

Sorting and searching

• Merge sort, bucket sort, quicksort, heapsort, binary search.

Graphs algorithms and data structures

• Graphs traversals, shortest paths, spanning trees, transitive closure, decompostitions, matching, network flows.

Complexity

• Polynomial reductions, NP-completeness.

Keywords

algorithms, data structures, efficiency, problem solving

Learning Prerequisites

Recommended courses

Discrete Structures

Learning Outcomes

By the end of the course, the student must be able to:

- Illustrate the execution of algorithms on example inputs
- Describe basic data structures such as arrays, lists, stacks, queues, binary search trees, heaps, and hash tables
- Analyze algorithm efficiency
- Compare alternative algorithms and data structures with respect to efficiency
- Choose which algorithm or data structure to use in different scenarios
- Use algorithms and data structures taught in the course on concrete problem instances
- Design new algorithms and data structures based on known methods
- Prove the correctness of an algorithm

Teaching methods

Ex cathedra lecture, exercises in classroom

Assessment methods

Continuous assessment with final exam.

Resources

Bibliography

Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein: *Introduction to algorithms*, Third Edition, MIT Press, 2009.

Ressources en bibliothèque

- Introduction to algorithms / Cormen

Websites

- http://theory.epfl.ch/courses/algorithms

MATH-203(c)

Analyse III

Nguyên Hoài-Minh

Cursus	Sem.	Type
HES - IN	H	Obl.
HES - SIE	H	Obl.
HES -SC	H	Obl.
Informatique	BA3	Obl.
Sciences et ingénierie de l'environnement	BA3	Obl.
Systèmes de communication	BA3	Obl.

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo

Résumé

Le cours étudie les concepts fondamentaux de l'analyse vectorielle et l'analyse de series de Fourier en vue de leur utilisation pour résoudre des problèmes pluridisciplinaires d'ingénierie scientifique.

Contenu

Le cours contient deux parties:
1) Analyse vectorielle
Les opérateurs gradient, rotationnel et divergence. Intégrales curvilignes et intégrales de surfaces. Champs vectoriels et potentiels. Théorèmes de Green, de la divergence et de Stokes.
2) Analyse de Fourier
Séries de Fourier. Identité de Parceval.

Compétences requises

Cours prérequis obligatoires

Analyse I, Analyse II, Algèbre linéaire.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Comprendre et maîtriser les notions, les concepts et les méthodes étudiés au cours.
- Comprendre et maîtriser les notions, les concepts et les méthodes pratiqués dans les séries d'exercices.

Méthode d'enseignement

Cours ex cathedra, exercices en salle.

Méthode d'évaluation

Examen écrit.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non

Ressources

Bibliographie

B. Dacorogna et C. Tanteri, *Analyse avancée pour ingénieurs,* PPUR 2011.

Ressources en bibliothèque

• Analyse avancée pour ingénieurs / Dacorogna

Préparation pour

Analyse IV.

MATH-207(d)   **Analyse IV**

Nguyên Hoài-Minh

Cursus	Sem.	Type
Génie civil	BA6	Opt.
HES - SIE	E	Obl.
HES -SC	E	Obl.
Informatique	BA4	Opt.
Sciences et ingénierie de l'environnement	BA4	Obl.
Systèmes de communication	BA4	Obl.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo

Résumé

Analyse complexe: fonctions holomorphes, équations de Cauchy-Riemann, intégration complexe, théorème de Cauchy, formule de Cauchy, séries de Laurent, théorème des résidus. Transformées de Fourier. Identité de Plancherel.

Contenu

Analyse complexe: fonctions holomorphes,
équations de Cauchy-Riemann, intégration complexe, théorème de Cauchy, formule de Cauchy, séries de Laurent, théorème des résidus. Transformées de Fourier. Identité de Plancherel.

Compétences requises

Cours prérequis obligatoires

Algèbre linéaire, Analyse I, II, III

Concepts importants à maîtriser

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Comprendre et maîtriser les notions, les concepts et les méthodes étudiés au cours
- Comprendre et maîtriser les notions, les concepts et les méthodes pratiqués en exercices

Méthode d'enseignement

Cours ex cathedra et exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Assistants   Oui

Ressources

Bibliographie

B. Dacorogna et C. Tanteri, Analyse avancée pour ingénieurs, PPUR, 2e édition.

S. D. Fisher, Complex Variables, Dover.
D. W. Kammler, A first course in Fourier analysis, Prentice Hall.
E. Kreyszig, Advanced engineering mathematics, Wiley.

Ressources en bibliothèque

- Advanced engineering mathematics / Kreyszig
- Analyse avancée pour ingénieurs / Dacorogna
- A first course in Fourier analysis / Kammler
- Complex Variables / Fisher

CS-209 **Architecture des systems-on-chip**

Ienne Paolo

Cursus	Sem.	Type
Computer engineering minor	E	Opt.
Informatique	BA4	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA4	Opt.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
TP	2 hebdo

Résumé

Le cours continue l'introduction des étudiants aux concepts de base de l'architecture des ordinateurs (entrée/sorties, interruptions et exceptions). Il aborde aussi le thème du parallélisme au niveau des instructions (pipelines, etc.) et les problèmes de la hiérarchie mémoire des multiprocesseurs.

Contenu

- Entrées/sorties et interruptions.
- Exceptions.
- Accès directe à la mémoire.
- Performance des ordinateurs.
- Pipelines.
- Processeurs à ordonnancement dynamique.
- Processeurs superscalaires et VLIW.
- Multiprocesseurs.

Mots-clés

Architecture des ordinateurs, Systèmes-on-chip, Processeurs, Interruptions et exceptions, Parallélisme au niveau des instructions, Multiprocesseurs.

Compétences requises

Cours prérequis obligatoires

- Conception de systèmes numériques.
- Architecture des ordinateurs.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Réaliser des gestionnaires d'exception simples en langage assembleur.
- Concevoir au niveau "Register Transfer Logic" des systèmes digitaux en pipeline.
- Optimiser la performance d'un pipeline en réordonnançant les opérations.
- Illustrer es difficultés liées à la construction de systèmes multiprocesseurs.

Méthode d'enseignement

Cours et travaux pratiques sur une carte FPGA dediée.

Méthode d'évaluation

Questionnaires à choix multiples, test intermédiaire et examen final.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kauffman, 5th edition, 2013.

Ressources en bibliothèque

- Computer Organization and Design / Patterson

Sites web

- http://fpga4u.epfl.ch

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=14153

Préparation pour

Advanced Computer Architecture.

CS-308   **Calcul quantique**

Macris Nicolas

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	3 hebdo
Exercices	1 hebdo

Remarque

Cours indépendant de "Traitement quantique de l'information" (COM-309)

Résumé

Avec la miniaturisation des ordinateurs nous sommes conduits à réviser les paradigmes du calcul classique pour développer des modèles de calcul quantique. Le cours introduit les notions de bit quantique, les portes logiques et circuits quantiques, puis traite les principaux algorithmes quantiques.

Contenu

Intrduction au calcul quantique
- Calcul classique: modèle des circuits classiques, calcul réversible.
- Bits quantiques, espace de Hilbert de N qubits, transformations unitaires et portes logiques élémentaires, postulat de la mesure.
- Modèle des circuits quantiques, portes universelles.
- Problème de Deutsch et Josza.
Algorithmes de base
- Sous espace vectoriel cache et algorithme de Simon.
- Intermède mathématique: factorisation d'un entier et période de fonctions discrètes. Notions sur les fractions continuées.
- Transformée de Fourier quantique et algorithme de recherche de la période d'une fonction discrète.
- Algorithme de factorisation de Shor.
- Algorithme de Grover pour la recherche dans une base de donnée.
Intrication (sujet a choix et facultatif)
- Etats intriques et circuits associés.
- Protocoles avec opérations locales quantiques + communication classique.
- Protocole de distillation et mesure de l'intrication.
Codage (sujet a choix et facultatif)
- Modèles de bruit et erreurs dans les états quantiques**.**
- Code correcteurs de Shor et Steane.
- Codes stabilisateurs.

Mots-clés

Calcul quantique, circuits quantiques, portes universelles, transformée de Fourier quantique, algorithme de Shor, Grover, intrication, codes quantiques.

Compétences requises

Cours prérequis obligatoires
Algèbre linéaire.

Concepts importants à maîtriser

Matrices, valeurs et vecteurs propres, produit scalaire, nombre complexes.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Expliquer le concept d'algorithme quantique sur le modèle des circuits.
- Connaitre les portes universelles utilisées dans un circuit quantique.
- Expliquer les principaux algorithmes quantiques
- Calculer l'évolution d'un état à travers un circuit quantique
- Appliquer le postulat de la mesure
- Faire des calculs algébriques impliquant des états à plusieurs qubits en notation de Dirac

Méthode d'enseignement

Ex-Cathedra. Exercices. Lectures d'articles pedagogiques sur les sujets a choix.

Travail attendu

Participation au cours, exercices et lectures pedagogiques sur les sujets a choix.

Méthode d'évaluation

midterm 20%, 4 graded homeworks 20%, examen final ecrit 60%.

Ressources

Bibliographie

N. David Mermin: *Quantum Computer Science, an introduction.* Cambridge University Press
Nielsen and Chuang: *Quantum Computation and Information.* Cambridge University Press

Ressources en bibliothèque

- Quantum Computation and Information / Nielsen
- Quantum Computer Science / Mermin

Polycopiés

Notes de cours

Sites web

- http://ipg.epfl.ch/doku.php?id=en:courses

Préparation pour

COM-611 Quantum Information Theory and Computation

EE-204   **Circuits et systèmes I**

Rachidi-Haeri Farhad

Cursus	Sem.	Type
Génie électrique et électronique	BA3	Obl.
HES - EL	H	Opt.
HES -SC	H	Obl.
Informatique	BA3	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA3	Obl.

Langue	français
Crédits	3
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Ce cours présente une introduction à la théorie et aux méthodes d'analyse et de résolution des circuits électriques.

Contenu

1. Concepts de base et lois fondamentales
- charge, courant, tension
- puissance et énergie
- éléments de circuits (R, L, C, sources indép. et commandées, inductances couplées)
- lois de Kirchhoff
- limitations des équations de Kirchhoff
- Combinaison d'éléments/méthodes de réduction de circuit
- Diviseurs de tension/courant
- Substitution de source

2. Théorèmes fondamentaux
- Théorème de Thévenin
- Théorème de Norton
- Transfert maximal de puissance
- Théorème de Superposition

3. Méthodes d'analyse
- Analyse nodale
- Analyse de maille

5. Analyse des circuits en régime sinusoïdal
- Phaseurs et calcul complexe
- Impédance et admittance
- Puissance active et réactive
- Théorèmes de Thévenin et Norton en régime sinusoïdal
- Transfert de Puissance active
- Optimisation pour la transmission d'information et pour la transmission d'énergie

6. Circuits en régime triphasé
- Systèmes triphasés symétriques
- Transformation triangle-étoile
- Puissances en régime triphasé
- Systèmes triphasés asymétriques

7. Régimes transitoires
- Réponse impulsionnelle et indicielle
- Méthode d'analyse
- Réponse indicielle des circuits du premier ordre, constantes de temps
- Réponse indicielle de circuits du second ordre

8. Quadripôles
- Paramètres d'impédance et d'admittance
- Paramètres hybrides
- Paramètres de transmission
- Relation entre paramètres

Compétences requises

Cours prérequis indicatifs

Analyse I et II, Algèbre linéaire

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Analyser un circuit électrique en régime permanent et transitoire.
- Analyser

Compétences transversales

- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Ex cathedra. Exercices sur papier.

Méthode d'évaluation

Examen écrit.

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

M. Jufer et Y. Perriard, Electrotechnique, PPUR, 2e édition, 2014.
C.K. Alexander et M.N.O. Sadiku, Analyse des circuits électriques, De Boeck, 2012
K.C.A. Smith and R.E. Alley, Electrical Circuits: An Introduction, Cambridge University Press, 1992.
James W. Nilsson et al.,"Electric Circuits". Prentice Hall, 7th ed., 2005.
Circuits et Systèmes, Martin Hasler, Notes de cours, Edition 1999, EPFL.
Support distribué au cours

Ressources en bibliothèque

- Analyse des circuits électriques / Alexander
- Electrotechnique / Jufer
- Electrical Circuits / Smith
- Signals and Systems / Oppenheim
- Electric Circuits / Nilsson
- Circuits et Systèmes / Hasler

Liens Moodle

- http://moodle.epfl.ch/enrol/index.php?id=14240

Préparation pour

Circuits et systèmes II, Filtres électriques, Automatique, Réseaux électriques

EE-205

Circuits and systems II

Gastpar Michael Christoph

Cursus	Sem.	Type
Génie électrique et électronique	BA4	Obl.
HES - EL	E	Opt.
HES -SC	E	Obl.
Informatique	BA4	Opt.
Mineur en Systèmes de communication	E	Opt.
Systèmes de communication	BA4	Obl.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

This class teaches the theory of linear time-invariant (LTI) systems. These systems serve both as models of physical reality (such as the wireless channel) and as engineered systems (such as electrical circuits, filters and control strategies).

Content

The design of advanced systems (such as WiFi, cell phones, drones, airplanes) requires a thorough theoretical underpinning. This class teaches one of the most powerful and important pillars: The theory of linear time-invariant (LTI) systems. These systems serve both as models of physical reality (such as the wireless channel) and as engineered systems (such as filters and control strategies).
The class will cover the following topics:

1. Systems: Definitions (1 week)
2. LTI Systems (3 weeks)
3. The Frequency Response of stable LTI Systems (1 week)
4. Fourier Techniques for stable LTI Systems (3 weeks); with applications to Communication Systems and Signal Processing
5. Laplace and Z-Transform Techniques for LTI Systems (5 weeks); with applications to Control Systems

Keywords

Systems, Circuits, Signals, Frequency Response, Transfer Function, Fourier Transform, Laplace Transform, Z Transform, Stability, Causality, Sampling

Learning Prerequisites

Required courses

Analysis I, II, III. Linear algebra I.

Recommended courses

Linear algebra II

Learning Outcomes

By the end of the course, the student must be able to:
- Describe properties of LTI systems
- Solve for poles and zeros of LTI systems
- Recall properties of CT Fourier transform
- Analyze LTI systems by spectral analysis

- Operate with Fourier transform tools
- Work out / Determine impulse response of CT LTI

Teaching methods

- Classroom lectures
- Written exercises
- Graded homework problems

Expected student activities

- Read course book in english (the course is taught in english)

Assessment methods

Homeworks and written mid-term exam and final exams

Resources

Bibliography

The following is a recommended (but not required) book:
A. V. Oppenheim and A. S. Willsky, with S. Hamid Nawab, Signals and Systems. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1996.

Ressources en bibliothèque

- Signals and Systems / Oppenheim

Notes/Handbook

will be made available

CS-208 | **Computer architecture**
Stojilovic Mirjana

Cursus	Sem.	Type
Computer engineering minor	H	Opt.
HES - IN	H	Obl.
Informatique	BA3	Obl.
Mineur en Informatique	H	Opt.
Systèmes de communication	BA3	Obl.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Practical work	2 weekly

Summary

The course introduces the students to the basic notions of computer architecture and, in particular, to the choices of the Instruction Set Architecture and to the memory hierarchy of modern systems.

Content

• Complex digital systems in VHDL.
• Basic components of a computer.
• Instruction Set Architectures.
• Assembly-level programming.
• Multi-cycle implementation of processors.
• Caches.
• Virtual memory.

Keywords

Computer Architecture, Basic Processor Architecture, Instructions Sets, Cache Hierarchies, Virtual Memory.

Learning Prerequisites

Required courses
• Conception de systèmes numériques

Learning Outcomes

By the end of the course, the student must be able to:
• Design and implement a processor at the Register Transfer Level using logic synthesizers and simulators.
• Develop assembly language programs.
• Justify the organization of a modern memory system including cache hierarchies and virtual memory.
• Design and implement a cache memory.
• Design and implement a virtual memory.

Teaching methods

• Ex cathedra lectures and exercises.
• Labs on a dedicated FPGA board.

Expected student activities

- Attending the course and exercise/lab sessions.
- Completing the lab assignments.
- Homework: solving the exercises in the course exercise book.
- Participating in the discussions on the forum.

Assessment methods

Continuous assessment.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kauffman, 5th edition, 2013.

Ressources en bibliothèque

- Computer Organization and Design: The Hardware-Software Interface / Patterson

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14225

Prerequisite for

- Architecture des systems-on-chip.

CS-320 **Computer language processing**

Kuncak Viktor

Cursus	Sem.	Type
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Practical work	2 weekly

Summary

We teach the fundamental aspects of analyzing and interpreting computer languages, including the techniques to build compilers. The new title is "Computer Language Processing".

Content

1. Overview, source languages and run-time models
2. Review of formal languages
3. Lexical analysis
4. Syntactic analysis (parsing)
5. Name analysis
6. Type checking
7. Code generation
8. Data-flow analysis
9. Run-time organization and memory management

Keywords

programming language;
compiler;
interpreter;
regular expression;
context-free grammar;
type system;
code generation;
static code analysis

Learning Prerequisites

Recommended courses

Discrete structures
Theoretical computer science
Programming in Scala
Computer architecture I

Learning Outcomes

By the end of the course, the student must be able to:

- Design a programming language
- Construct a compiler
- Coordinate development with project partner
- Formulate correctness conditions for compiler
- Estimate time to implement a programming language feature
- Produce a working programming language implementation
- Decide which language features make implementation difficult
- Specify programming language and compiler functionality

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Respect the rules of the institution in which you are working.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate a capacity for creativity.
- Take feedback (critique) and respond in an appropriate manner.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

- Ex catedra
- Exercises on whiteboard
- Exercises using dedicated software
- Project work, indepdently and under supervision of assistants

Assessment methods

- 50% Project
- 25% Mid-term quiz
- 25% End-of-term quiz in December

Resources

Bibliography

Andrew W. Appel, **Modern compiler implementation in Java** (or **ML**), Addison-Wesley 1997 (full PDF available from EPFL library)
Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: **Compilers: Principles, Techniques, and Tools** (2nd Edition, 2006)
Niklaus Wirth: **Compiler Construction,** neat textbook from a prominent classical authority. Freely available
 http://www.ethoberon.ethz.ch/WirthPubl/CBEAll.pdf

Ressources en bibliothèque

- Additionally, all material
- Modern compiler implementation in Java / Appel
- Compiler Construction / Wirth
- Compilers, principle, techniques and tools / Aho

Notes/Handbook

http://lara.epfl.ch/w/cc
Faboulous and gently paced videos: https://www.coursera.org/course/compilers

Prerequisite for

Synthesis, analysis and verification
Advanced compiler construction
Recommended for Foundations of software

COM-208

Computer networks

Argyraki Aikaterini

Cursus	Sem.	Type
HES - IN	H	Obl.
HES -SC	H	Obl.
Information security minor	H	Opt.
Informatique	BA3	Obl.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA3	Obl.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	During the semester
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

This course provides an introduction to computer networks. It describes the principles that underly modern network operation and illustrates them using the Internet as an example.

Content

- Overview of Internet operation (main components and protocols).
- Application layer (web, cookies, ads, email, peer to peer).
- Socket programming (how to write a very simple network application).
- Transport layer (UDP, TCP, congestion control).
- Network layer (IP forwarding and basic routing).
- Data link layer (switching and basic shared access protocols).
- Security (secure email, SSL, IPsec).

Keywords

- Computer networks
- Internet
- HTTP
- Peer-to-peer networks
- Sockets, TCP/IP, congestion control, routing, switching, network security.

Learning Prerequisites

Required courses

- CS 106 - Introduction to programming
- COM 101 - Information sciences

Learning Outcomes

By the end of the course, the student must be able to:
- Design simple network applications.

- Choose which functions to implement at each network layer.
- Compare different network protocols.
- Perform simple network troubleshooting.
- Use simple network monitoring tools.
- Implement simple client-server applications.
- Investigate simple network attacks.
- Explain how basic Internet applications work.
- Explain how TCP/IP works.

Teaching methods

- Lectures
- Reading sssignments
- Homework problems
- Hands-on exercises

Expected student activities

The students are expected to:

- attend the lectures
- read the assigned book sections
- complete homework problems
- complete hands-on exercises.

Assessment methods

- Quizzes and short essay (bonus points that can contribute up to 10% of the grade).
- Midterm exam (40% of the grade).
- Final exam (60% of the grade).

Supervision

Office hours   Yes
Assistants   Yes
Forum   No

Resources

Bibliography

Computer Networking: A Top-Down Approach by James F. Kurose and Keith W. Ross.

Ressources en bibliothèque

- Computer Networking / Kurose

Websites

- http://compnet.epfl.ch

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

EE-200 — Electromagnétisme I : lignes et ondes

Fleury Romain Christophe Rémy

Cursus	Sem.	Type
Génie électrique et électronique	BA3	Obl.
HES - EL	H	Obl.
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Langue	français
Crédits	3
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Le signal électrique est un vecteur essentiel pour la transmission d'information et d'énergie. En haute fréquence elle se manifeste comme un signal électromagnétique dont l'étude demande le développement de modèles physiques et mathématiques spécifiques basés sur les équations d'onde.

Contenu

1) Le signal électromagnétique
Aspects spécifiques du signal électromagnétique: Signaux scalaires et vectoriels. Signaux guidés et rayonnés. Domaines temporels et fréquentiel. Affaiblissement, dispersion et distorsion. Puissance transmise et vecteur de Poynting.

2) Lignes de transmission et circuits HF
Dimensions du circuit, fréquence et longueur d'onde. Eléments discrets (localisés) et distribués. Circuits à un et à plusieurs accès, éléments réciproques et sans pertes, bilan de puissance. Matrice de répartition d'un quadripôle. Vitesses de phase et de groupe, impédance caractéristique, réflexion et transmission, ondes stationnaires, transfert de puissance et méthodes d'adaptation. Abaque de Smith

3) Propagation d'ondes
Analogie avec la théorie des lignes de transmission. Equations de Maxwell. Polarisation linéaire, circulaire et elliptique. Incidence normale et oblique sur un obstacle plan. Réflexion et transmission. Diffraction. Étude de cas particuliers. Milieux dissipatifs et profondeur de peau.

Mots-clés

Signal électromagnétique, Lignes de transmission, Ondes électromagnétiques, Réflexion et transmission, Circuits équivalents, Circuits radiofréquences, Propagation et Rayonnement, Antennes

Compétences requises

Cours prérequis obligatoires
Electrotechnique

Cours prérequis indicatifs
Algèbre, Analyse I et II, Physique générale

Concepts importants à maîtriser
Propriétés du signal électromagnétique: vitesse, fréquence, longueur d'onde. Nature et comportement des signaux et ondes électromagnétiques: propagation guidée unidimensionnelle (lignes de transmission), champs électromagnétiques, polarisation, interaction avec la matière, réflexion et transmission, rayonnement

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Analyser un système des lignes de transmission
- Calculer les impédances et paramètres électriques associés d'un circuit radiofréquence
- Modéliser une interaction simple entre onde électromagnétique et matière

Méthode d'enseignement

Ex cathedra avec exercices en salle. Les exercices peuvent nécessiter l'utilisation d'un ordinateur (MatLab).

Travail attendu

Participation active au cours et aux séances d'exercices.

Méthode d'évaluation

Examen écrit.

Encadrement

Office hours   Oui
Assistants   Oui

Ressources

Bibliographie

1) "Électromagnétisme", Vol. III du Traité d'électricité de l'EPFL - 2) Ramo: "Fields and Waves in Communication Electronics" - 3) Notes supplémentaires polycopiées

Ressources en bibliothèque

- Fields and Waves in Communication Electronics / Ramo
- Electromagnétisme / Gardiol

Liens Moodle

- http://moodle.epfl.ch/enrol/index.php?id=14046

Préparation pour

Transmissions Hyperfréquences et Optiques, Télécommunications, Rayonnement et Antennes, Propagation, Audio, cycle Master EPFL-SEL et EPFL-SC

EE-201   **Electromagnétisme II : calcul des champs**

Fleury Romain Christophe Rémy

Cursus	Sem.	Type
Génie électrique et electronique	BA4	Obl.
HES - EL	E	Obl.
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Ce cours traite de l'électromagnétisme dans le vide et dans les milieux continus. A partir des principes fondamentaux de l'électromagnétisme, on établit les méthodes de résolution des équations de Maxwell dans le vide et dans des milieux matériels complexes.

Contenu

1) Théorèmes fondamentaux de l'Electromagnétisme
Equations de Maxwell. Conservation de la charge. Théorème de Poynting. Dualité. Réciprocité de Lorentz. Unicité du champ électromagnétique.
2) Calcul des champs rayonnés
Potentiels vecteurs et scalaires. Champs émis par une distribution arbitraire de courant. Le dipole infinitésimal. Fonction de Green. Champ proche et champ lointain. Méthode de calcul direct du champ lointain. Principe de Huygens. Méthode des images et applications.
3) Electrodynamique des milieux continus
Interaction avec la matière. Relations constitutives. Dispersion spatiale et fréquentielle. Linéarité. Permittivité et perméabilité. Milieux bi-anisotropes et uniaxiaux. Conservation de l'énergie et réciprocité. Solution des équations de Maxwell dans des milieux complexes. Relation de dispersion. Modèle de Drude-Lorentz. Diélectriques et métaux aux fréquences optiques. Matériaux plasmoniques.

Mots-clés

electromagnetisme, distributions de charges et courants électriques, champs électriques et magnétiques, rayonnement, champ lointain, milieux continus

Compétences requises

Cours prérequis obligatoires
Physique Générale

Cours prérequis indicatifs
Analyse I et II, Electrotechnique

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Utiliser les théorèmes fondamentaux de l'électromagnétisme
- Déterminer les propriétés électromagnétiques d'un milieu à partir des relations constitutives
- Calculer les champs électriques et magnétiques rayonnés par une distribution de charge

Méthode d'enseignement

Ex cathedra avec exercices en salle. Certains exemples pourront être traités à l'ordinateur (MatLab).

Travail attendu

Participation active au cours et aux séances d'exercices.

Méthode d'évaluation

Examen écrit.

Encadrement

Office hours   Oui
Assistants   Oui

Ressources

Bibliographie

1) "Électromagnétisme", Vol. III du Traité d'électricité de l'EPFL
2) Ramo: "Fields and Waves in Communication Electronics"
3) Notes supplémentaires polycopiées

Ressources en bibliothèque

• Electromagnétisme / Gardiol
• Fields and Waves in Communication Electronics / Ramo

Polycopiés

Disponible sur Moodle

Liens Moodle

• http://moodle.epfl.ch/enrol/index.php?id=14047

Préparation pour

Transmissions Hyperfréquences et Optiques, Télécommunications, Orientation Communications mobiles, Rayonnement et Antennes, Propagation, Audio

EE-202(b)   **Electronique I**
Zysman Eytan

Cursus	Sem.	Type
Informatique	BA3	Opt.
Systèmes de communication	BA3	Opt.

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Découvrir le monde de l'électronique depuis les lois fondamentales des composants discrets linéaires et non linéaires. Les circuits obtenus avec des assemblages de composants nécessitent de nombreuses techniques de modélisation et d'analyse ainsi que des vérification exploitant un simulateur

Contenu

Cours

- Composants passifs linéaires
- Techniques de résolution de circuits linéaires
- Les diodes
- introduction aux transistors
- Techniques de modélisation des composants non linéaires
- Simulation électronique

Exercices

L'étudiant appliquera les nombreuses méthodes vues en cours pour résoudre des exercices pratiques qui pourront être vérifiés avec la simulation.

Mots-clés

Composants passifs, composants actifs, composants linéaires, composants non linéaires, diodes, transistors, modélisation, simulation, Lois de Kirchhoff, Thévenin-Norton, Superposition, impédances complexes, fonctions de transfert, Bode, concept d'amplification.

Compétences requises

Cours prérequis obligatoires

Cours d'analyse: équation différentielles du premier et second ordre, nombres complexes, résolution de système d'équations linéaires.

Cours prérequis indicatifs

Electricité de base: électrostatique, électrocinétique.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Analyser des circuits complexes

- Modéliser des composants non linéaires
- Modéliser des circuits complexes
- Raisonner à partir de méthode d'observation
- Dessiner des comportements temporels et fréquentiels
- Interpréter des signaux de natures diverses
- Utiliser les bonnes méthodes de résolution

Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.

Méthode d'enseignement

Cours ex cathedra et exercices dirigés en salle.

Méthode d'évaluation

Plusieurs quiz répartis sur le semestre
Travail écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

- **Principes d'électronique: cours et exercices corrigés. Albert Paul Malvino ; trad. de l'américain par Bernard Boittiaux ; Paris : Dunod, 2002**

Ressources en bibliothèque

- Principes d'électronique / Malvino

Polycopiés

- liste de sites approfondissant les notions vues en cours
- Diapositives du cours
- Diapositives commentées
- Exercices et corrigés.
- Développements en cours sur Tablet

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=13726

Préparation pour

Électronique II

EE-203(b)   **Electronique II**

Zysman Eytan

Cursus	Sem.	Type
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo

Résumé

Maîtriser des blocs fonctionnels nécessitant un plus haut niveau d'abstraction. Réalisation de fonctions électroniques de haut niveau exploitant les amplificateurs opérationnels.

Contenu

Cours

- Modèles d'amplificateurs
- Bande passante des amplificateurs
- Familles logiques
- l'amplificateur opérationnel en réaction négative
- l'amplificateur opérationnel en réaction positive
- les filtres actifs d'ordre N
- l'amplificateur opérationnel et ses imperfections
- les bascules

Exercices et travaux pratiques

Comme en électronique I, l'étudiant appliquera de nombreuses méthodes vues en cours pour résoudre des exercices pratiques qui pourront être vérifiés avec la simulation.

Mots-clés

Amplificateur, Modèle de quadripôle, polarisation, schéma petit signaux, Filtres, bande passante, puissante statique, puissance dynamique, Slew-rate, Tchebychev, Butterworth, Trigger de Schmitt, comparateur, intégrateur, différentiateur, monostable, bistable, astable, générateur de signaux, marge de bruit, Fan-In, Fan-Out, Puissance dissipée, tension d'offset.

Compétences requises

Cours prérequis indicatifs

Électronique I

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Concevoir des filtres actifs
- Concevoir des circuits amplificateurs
- Comparer les différentes familles logiques

- Analyser la bande passante d'une fonction électronique
- Exploiter des blocs fonctionnels de haut niveau
- Représenter la notion de temps
- Synthétiser des circuits logiques

Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.

Méthode d'enseignement

Cours ex cathedra et exercices dirigés en salle.

Méthode d'évaluation

Plusieurs quiz répartis sur le semestre
Travail écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

Principes d'électronique: cours et exercices corrigés. Albert Paul Malvino ; trad. de l'américain par Bernard Boittiaux ; Paris : Dunod, 2002

Ressources en bibliothèque

- Principes d'électronique / Malvino

Polycopiés

- liste de sites approfondissant les notions vues en cours
- Diapositives du cours
- Diapositives commentées
- Exercices et corrigés.
- Développements en cours sur Tablet

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=13727

Préparation pour

Electronique III

EE-381   **Electronique III**

Zysman Eytan

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Comparaison entre les systèmes à composants discrets et les systèmes intégrés. Introduction aux systèmes électroniques numériques et analogiques et à leur interfaçage. Analyse sous forme d'un projet d'un cahier des charges d'un système intégré mixte analogique/numérique.

Contenu

Cours

- Cellules analogiques: miroir de courant, paire différentielle, Push-pull,...
- Conversion A/N et N/A : introduction - définitions, conversion numérique/analogique, conversion analogique/numérique.
- Oscillateur et boucles à verrouillage de phase ou Phase-Locked Loops (PLL)
- Introduction aux technologies mixtes analogiques et numériques
- Techniques de conception de circuits intégrés
- Application aux ASIC analogiques/numériques

Exercices
l'étudiant analysera et simulera de nombreux blocs fonctionnels vus en cours
projet
L'étudiant fera la conception d'un petit système électronique mixte analogique et numérique et évaluera sa complexité sous forme de circuit intégré.

Mots-clés

paire différentielle, miroir de courant, structure cascod, charge active, Push-Pull, Darlington, Wilson, Widlar, Full Custom, Semi-custom, Librairie de cellules, FPGA, EPLD, PLA, ROM, Architecture de circuit intégré, Technologie des semiconducteurs, PLL, Stabilité, Oscillateur, Convertisseur incrémental, convertisseur logarithmique, convertisseur flash et semi Flash, Sigma/Delta.

Compétences requises

Cours prérequis indicatifs

Cours d'électronique de base I et II

Concepts importants à maîtriser

Automates de Moore et de Mealy.
Transformée de Laplace.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Analyser un cahier des charges en électronique
- Concevoir un système électronique
- Décrire le comportement du circuit sous forme algorithmique
- Estimer la complexité et les performances du circuit

Compétences transversales

- Fixer des objectifs et concevoir un plan d'action pour les atteindre.
- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Accéder aux sources d'informations appropriées et les évaluer.
- Ecrire un rapport scientifique ou technique.
- Faire une présentation orale.

Méthode d'enseignement

- Cours ex cathedra et exercices dirigés en salle.
- Animation de séances de projet

Travail attendu

- Remise d'un rapport d'analyse de système électronique

Méthode d'évaluation

- Travail écrit
- Rapport et présentation orale du projet

Encadrement

Office hours	Non
Assistants	Non
Forum électronique	Oui

Ressources

Bibliographie

Traité de l'électronique analogique et numérique , 1, Techniques analogique et numérique, Paul Horowitz, Winfield Hill, Elektor, 2009

Ressources en bibliothèque

- Traité de l'électronique analogique et numérique (vo.1)/ Horrowitz

Polycopiés

- liste de sites approfondissant les notions vues en cours

- Diapositives du cours
- Diapositives commentées
- Exercices et corrigés.
- Développements en cours sur Tablet
- Cahier des charges du projet

CS-210   **Functional programming**

Kuncak Viktor, Odersky Martin

Cursus	Sem.	Type
HES - IN	H	Obl.
Informatique	BA3	Obl.
Systèmes de communication	BA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	During the semester
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

Understanding of the principles and applications of declaratative programming, the fundamental models of program execution, application of fundamental methods of program composition,meta-programming through the construction of interpreters and advanced programming techniques.

Content

Introduction to programming in Scala
Expressions and functions
Classes and objects
Evaluation by rewriting
Pattern matching
Polymorphism
Evaluation strategies
Domain-specific languages
Constraint programming
Language interpretation
An interpreter for Lisp
An interpreter for Prolog

Learning Prerequisites

Required courses

Introduction to the programming objet
Theory and practice of programming

Important concepts to start the course

Compiler Construction
Foundations of Software

Learning Outcomes

By the end of the course, the student must be able to:

- Create functional programs

- Design robust and readable software

- Formalize program correctness

- Interpret programs automatically

- Prove correctness using induction

- Construct software

Transversal skills

- Demonstrate a capacity for creativity.
- Use a work methodology appropriate to the task.
- Set objectives and design an action plan to reach those objectives.
- Give feedback (critique) in an appropriate fashion.

Teaching methods

MOOC. Ex Cathedra. Exercises and projects

Assessment methods

Continuous and written test at the end of the course

Resources

Bibliography

Abelson/Sussman : Structure and Interpretation of Computer Programs, MIT Press

Ressources en bibliothèque

- Structure and Interpretation of Computer Programs / Abelson

Websites

- http://Lampwww.epfl.ch/teaching

PHYS-114   **General physics II**
Dil Jan Hugo

Cursus	Sem.	Type
HES - IN	H	Obl.
HES -SC	H	Obl.
Informatique	BA3	Obl.
Systèmes de communication	BA3	Obl.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

The course first develops the basic laws of electricity and magnetism and illustrates the use in understanding various electromagnetic phenomena.

Content

ELECTRICITY AND MAGNETISM
Electric fields: electric charges and fields; Coulomb's law; Gauss's law
Electric potential and energy: potential; energy; capacitance and capacitors; dielectric materials
Magnetism: magnetic forces and fields; Ampere's law; Biot-Savart law
Electromagnetism: electromotive force; Farady's law; inductance and inductors; Maxwell's equations
Electromagnetic waves: electromagnetic spectrum; antennas

Learning Prerequisites

Recommended courses

General Physics I

Learning Outcomes

By the end of the course, the student must be able to:
- Formulate approach for solving physics problems
- Analyze physical systems
- Establish competence in complex problem solving

Transversal skills

- Use a work methodology appropriate to the task.
- Take feedback (critique) and respond in an appropriate manner.
- Access and evaluate appropriate sources of information.

Teaching methods

Ex cathedra with demonstrations, exercises in class

Assessment methods

only final written exam

Supervision

Assistants   Yes

Resources

 Bibliography
 polycopiés / course notes

CS-321   **Informatique du temps réel**

Decotignie Jean-Dominique

Cursus	Sem.	Type
Informatique	BA3	Opt.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA3	Opt.

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	3 hebdo
Projet	1 hebdo

Résumé

A l'issue du cours, l'étudiant aura acquis les connaissances principales liées à la conception et à la réalisation des systèmes temps réel. Les différentes notions seront illustrées par des exercices et des laboratoires.

Contenu

1. Introduction sur l'informatique du temps-réel et ses particularités
2. Modélisation des systèmes temps-réel - contexte, types
3. Modélisation asynchrone du comportement logique - Réseaux de Petri
4. Modélisation des systèmes temps-réels - GRAFCET
5. Types de programmation (polling, par interruption, par états, exécutifs cycliques, coroutines, tâches)
6. Noyaux et systèmes d'exploitation temps-réel - problèmes, principes, mécanismes (tâches synchrones et asynchrones, synchronisation des tâches, gestion du temps et des événements)
7. Ordonnancement - problèmes, contraintes, nomenclature
8. Ordonnancement à priorités statiques (Rate Monotonic) et selon les échéances (EDF)
9. Ordonnancement en tenant compte des ressources, des relations de précédence et des surcharges
10. Ordonnancement de tâches multimédia
11. Evaluation des temps d'exécution
12. Introduction aux systèmes répartis temps réel

Mots-clés

temps réel, systèmes embarqués, systèmes enfouis, noyaux, ordonnancement, modélisation, GRAFCET, réseaux de Petri.

Compétences requises

Cours prérequis obligatoires
aucun

Cours prérequis indicatifs
Programmation

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- maîtrise des différentes techniques d'ordonnancement
- capacité de vérifier la garantie des contraintes temporelles
- modélisation des applications et vérifications de leurs proporiétés
- large connaissance de l'architecture des systèmes embarqués

Méthode d'enseignement

Ex cathedra + laboratoires + exercices

Travail attendu

apprentissage de la matière du cours, résoudre les exercices, réalisation des expériences pratiques

Méthode d'évaluation

Examen final 100%

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

G. Buttazzo, "Hard Real-Time Computing Systems", Kluwer Academic, Boston
P. Laplante, "Real-Time Systems Design & Analysis : An Engineer's Handbook", IEEE
R. David, A. Alla, "Petri nets and Grafcet", Prentice Hall

Ressources en bibliothèque

- Petri nets and Grafcet / David
- Hard Real-Time Computing Systems / Buttazzo
- Real-Time Systems Design & Analysis / Laplante

Sites web

- http://moodle.epfl.ch
- http://lamspeople.epfl.ch/decotignie/#InfoTR

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=10751

Préparation pour

Embedded systems, Real-time embedded systems, Real-time networks

CS-330 **Intelligence artificielle**

Faltings Boi

Cursus	Sem.	Type
Informatique	BA6	Opt.
Mineur en Informatique	E	Opt.
Sciences et technologies du vivant	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo

Résumé

Introduction aux techniques de l'Intelligence Artificielle, complémentée par des exercices de programmation qui montrent les algorithmes et des exemples de leur application à des problèmes pratiques.

Contenu

Le cours comporte trois segments qui traitent les 3 différents formes d'inférence logique : déduction, abduction et induction :

1. Représentation de connaissances en logique de prédicats, algorithmes d'inférence
2. Systèmes experts
3. Raisonnement imprécis et incertain
4. Algorithmes de recherche
5. Satisfaction de Contraintes
6. Diagnostic et Planification
7. Apprentissage supervisé
8. Apprentissage non-supervisé
9. Apprentissage bio-inspiré

Compétences requises

Cours prérequis indicatifs

Programmation avancée

Concepts importants à maîtriser

Logique de prédicats
Algorithmes de base
Théorie de probabilités
Programmation

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

• Choisir le bon type d'inférence pour une application

• Choisir la méthode la plus appropriée pour un certain type d'inférence

• Evaluer la faisabilité d'une application de l'Intelligence Artificielle

• Choisir, implémenter et décrire des algorithmes d'inférence déductive sur la base de calcul de prédicats

• Formuler des connaissances utilisant la logique des prédicats

- Décrire des méthodes d'inférence avec des informations imprécises et incertaines
- Choisir, implémenter et décrire des algorithmes de recherche et de satisfaction de contraintes
- Choisir et décrire des méthodes pour le diagnostic
- Choisir, implémenter et décrire des méthodes pour la planification
- Choisir, implémenter et décrire des méthodes d'apprentissage supervisé sur la base d'exemples
- Choisir, implémenter et décrire des méthodes d'apprentissage non-supervisé

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateur

Travail attendu

Participation au cours et exercices: 4 heures/semaine
Lecture: 2 heures/semaine
Travail independant: 3 heures/semaine

Méthode d'évaluation

Exercices 30%, examens intermediaire et final 70%

Ressources

Bibliographie

Boi Faltings, Michael Schumacher : Intelligence Artificielle par la pratique, PPUR
(Russel & Norvig : Artificial Intelligence : A Modern Approach / Prentice Hall)

Ressources en bibliothèque

- Intelligence Artificielle par la pratique / Faltings
- Artificial Intelligence / Russell

Sites web

- http://liawww.epfl.ch/
- http://moodle.epfl.ch/

Préparation pour

Intelligent Agents

COM-308

Internet analytics

Grossglauser Matthias

Cursus	Sem.	Type
Informatique	BA6	Opt.
Mineur en Systèmes de communication	E	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	2 weekly

Summary

Internet analytics is the collection, modeling, and analysis of user data in large-scale online services, such as social networking, e-commerce, search, and advertisement. This class explores a number of the key functions of such online services that have become ubiquitous over the past decade.

Content

The class seeks a balance between foundational but relatively basic material in algorithms, statistics, graph theory and related fields, with real-world applications inspired by the current practice of internet and cloud services.
Specifically, we look at social & information networks, recommender systems, clustering and community detection, search/retrieval/topic models, dimensionality reduction, stream computing, and online ad auctions. Together, these provide a good coverage of the main uses for data mining and analytics applications in social networking, e-commerce, social media, etc.
The course is combination of theoretical materials and weekly laboratory sessions, where we explore several large-scale datasets from the real world. For this, you will work with a dedicated infrastructure based on Hadoop & Apache Spark.

Keywords

data mining; machine learning; social networking; map-reduce; hadoop; recommender systems; clustering; community detection; topic models; information retrieval; stream computing; ad auctions

Learning Prerequisites

Required courses

Stochastic models in communication (COM-300)

Recommended courses

Basic linear algebra
Algorithms & data structures

Important concepts to start the course

Graphs; linear algebra; Markov chains; Java

Learning Outcomes

By the end of the course, the student must be able to:
- Explore real-world data from online services

- Develop frameworks and models for typical data mining problems in online services
- Analyze the efficiency and effectiveness of these models
- data-mining and machine learning techniques to concrete real-world problems

Teaching methods

Ex cathedra + homeworks + lab sessions

Expected student activities

Lectures with associated homeworks explore the basic models and fundamental concepts. The labs are designed to explore very practical questions based on a number of large-scale real-world datasets we have curated for the class. The labs draw on knowledge acquired in the lectures, but are hands-on and self-contained.

Assessment methods

Project 20%, midterm 30%, final exam 50%

Resources

Bibliography

C. Bishop, Pattern Recognition and MachineLearning, Springer, 2006
A. Rajaraman, J. D. Ullman: Mining of Massive Datasets, 2012
M. Chiang: Networked Life, Cambridge, Cambridge, 2012
D. Easley, J. Kleinberg: Networks, Crowds, and Markets, Cambridge, 2010
Ch. D. Manning, P. Raghavan, H. Schütze: Introduction to Information Retrieval, Cambridge, 2008
M.E.J. Newman: Networks: An Introduction, Oxford, 2010

Websites

- http://icawww1.epfl.ch/ix/

CS-211   Introduction à l'informatique visuelle

Dillenbourg Pierre, Salzmann Mathieu

Cursus	Sem.	Type
Bioingénierie	MA2, MA4	Opt.
Informatique	BA4	Opt.
Sciences du vivant	MA2, MA4	Opt.
Systèmes de communication	BA4	Opt.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Projet	2 hebdo

Résumé

Ce cours constitue le point d'entrée du thème 'visual computing' qui se continue dans le curriculum en informatique au bachelor et au master. Il explore le rôle des images, statiques ou animées, dans l'interaction entre un ordinateur, les utilisateurs et leur environnement.

Contenu

1. *Introduction à la perception humaine: Mécanismes de vision et de cognition.*
2. *Introduction à la formation des images: Projection d'une scène via une caméra, transformations de la caméra, modélisation et rendu 3D.*
3. *Introduction au traitement des images: Détection de contours et de formes, réalité augmentée.*
4. *Introduction à l'interaction personne-machine (HCI): Styles d'intéraction, principes de design, mesures de 'usability' et conception d'expériences.*
5. *Introduction aux mécanismes de jeu et théorie du 'flow'.*

Mots-clés

Informatique visuelle, traitement de l'image, vision par ordinateur, infographie, visualisation, jeux

Compétences requises

Cours prérequis obligatoires

Programmation en JAVA

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Expliquer les effets d'images en termes de processus cognitfs
- Réaliser un jeu de réalité agumentée
- Décrire un dispositif expérimental
- Expliquer les erreurs d'un utilisateur
- Critiquer une interface
- Appliquer des algorithmes d'infographie
- Appliquer des algorithmes de vision par ordinateur

Méthode d'enseignement

Cours structuré autour d'un projet en équipe de 3. Il porte sur le développement d'un jeu de réalité augmentée. Les cours ex-cathedra comprennent des petites expériences online sur les processus cognitifs.

Travail attendu

Le projet comprendra plusieurs étapes de rendu au cours du semestre

Méthode d'évaluation

- Projet + Exam
- 50 / 50

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=14247

BIO-109   **Introduction aux sciences du vivant (pour IC)**

Zufferey Romain

Cursus	Sem.	Type
Informatique	BA6	Opt.
Mineur en Biocomputing	E	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	6
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

Ce cours présente les principes fondamentaux à l'œuvre dans les organismes vivants. Autant que possible, l'accent est mis sur les contributions de l'Informatique aux progrès des Sciences de la Vie.

Contenu

Le cours aborde la plupart des concepts fondamentaux des Sciences de la Vie.
Les sujets développés parce qu'ils sont à l'interface avec l'informatique incluent :
- alignement des séquences, assemblage de séquences en génome
- matrice de distances et déduction d'un arbre phylogénétique
- détection de domaines transmembranaires et de signaux de localisation subcellulaire dans une séquence d'acides aminés.
- compositon en bases d'un génome entier, deuxième loi de parité de Chargaff, variations locales de la densité en CpG
- optimisation des codons dans diverses application pratiques.

Mots-clés

Bioinformatique, génome, séquençage, évolution, communication intercellulaires

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Identifier les principales structures cellulaires et comprendre les méthodes utilisées pour les observer
- Identifier les segments informatifs d'un génome
- Appliquer des algorithmes pour résoudre des questions en relation avec les Sciences de la Vie
- Expliquer le processus de l'expression génique
- Analyser des données expérimentales brutes et en tirer des conclusions sensées

Compétences transversales

- Accéder aux sources d'informations appropriées et les évaluer.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.

Travail attendu

En plus de la participations active aux cours et aux exercices, 4 heures de travail personnel sont attendues.

Méthode d'évaluation

Examen écrit durant la session d'été.

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Ressources

Polycopiés

Les dias du cours et les séries d'exercices sont mises à disposition du Moodle.

CS-341　　　**Introduction to computer graphics**

Pauly Mark

Cursus	Sem.	Type
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	BA6	Opt.
Mineur en Informatique	E	Opt.
Mineur en Systèmes de communication	E	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	2 weekly

Summary

The students study and apply fundamental concepts and algorithms of computer graphics for rendering, geometry synthesis, and animation. They design and implement their own interactive graphics programs.

Content

This course provides an introduction to the field of Computer Graphics. We will cover elementary rendering algorithms such as rasterization and raytracing, examine mathematical concepts and algorithms for geometric modeling, and then study concepts and algorithms for computer animation. Students will experiment with modern graphics programming and build small interactive demos. Complemented by some theoretical exercises, these programming tasks lead to a graphics software project, where small teams of students design and implement a complete graphics application.

Keywords

Pixels and images, 2D and 3D transformations, perspective transformations and visibility, rasterization, interpolation and lighting, OpenGL graphics API, raytracing, shader programming, texture mapping, procedural modeling, curves and surfaces, polygonal meshes, particle systems

Learning Prerequisites

Required courses
Nothing

Recommended courses
Linear Algebra

Learning Outcomes

By the end of the course, the student must be able to:
- Explain and apply the fundamental mathematical concepts of computer-based image and geometry synthesis
- Implement a basic rendering pipeline based on rasterization and raytracing
- Design and implement geometry synthesis based on procedural modeling
- Design and implement basic computer animation algorithms
- Integrate individual components into a complete graphics application
- Coordinate a team during a software project

Teaching methods

Lectures, interactive demos, theory and programming exercises, programming project, project tutoring

Expected student activities

The student are expected to study the provided reading material and actively participate in class. They should prepare and resolve the exercises, prepare and carry out the programming project. Exercises and project are done in groups of three students.

Assessment methods

Exercises and Project, Final Examination

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

A list of books will be provided at the beginning of the class

Ressources en bibliothèque

• Polygon mesh processing / Botsch

Notes/Handbook

Slides and online resources will be provided in class

Websites

• http://lgg.epfl.ch/ICG

Prerequisite for

Advanced Computer Graphics

CS-322   Introduction to database systems

Ailamaki Anastasia, Koch Christoph

Cursus	Sem.	Type
Energie et durabilité	MA2, MA4	Opt.
HES - IN	E	Obl.
Informatique	BA6	Obl.
Mineur STAS Russie	E	Opt.
Mineur en Informatique	E	Opt.
Sciences et ingénierie de l'environnement	MA2, MA4	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly

Summary

This course provides a deep understanding of the concepts behind data management systems. It covers fundamental data management topics such as system architecture, data models, query processing and optimization, database design, storage organization, and transaction management.

Content

This course provides a deep understanding of the concepts behind data management systems.
During this course, the students will learn about:

- The Entity-relationship and Relational Models
- Relational Algebra and Calculus
- The SQL Query Language
- Traditional and Modern Data Storage, File Organizations, and Indexing
- Hashing and Sorting
- Query Evaluation and Relational Operators
- Query Optimization
- Schema Refinement
- Transaction Management (Concurrency Control and Recovery)

Homework

Homeworks will be assigned to aid and assess comprehension of the above material. Homework will be either done using pen and paper or they will be programming exercises. During the semester the students will be asked to do a project to gain experience on how to build a database application, and to apply what they learn in class.

Keywords

databases, database design, data modeling, normalization, database management systems (DBMS), files, indexes, storage, external sorting, queries, query evaluation, query optimization, transactions, concurrency, recovery, SQL

Learning Prerequisites

Required courses

Data structures

Recommended courses

For the practical part of the course (project) the following skills will be needed:

• System oriented programming, with focus on scripting languages to enhance the parsing process of raw data.
• Building user interfaces, either web (e.g., PHP, JSP, ASP, ...) or application GUI (e.g., java).

Important concepts to start the course
Before the beginning of the course students must be familiar with:
• Data structures
• Algorithms concepts

Learning Outcomes

By the end of the course, the student must be able to:
• Express application information requirements
• Use a relational DBMS
• Create a database on a relational DBMS
• Design a database with a practical application in mind
• Model the data of an application using ER and relational modeling
• Explore how a DBMS performs its work
• Report performance and possible optimizations for applications using DBMS
• Justify design and implementation choices

Transversal skills

• Assess progress against the plan, and adapt the plan as appropriate.
• Evaluate one's own performance in the team, receive and respond appropriately to feedback.
• Write a scientific or technical report.
• Make an oral presentation.

Teaching methods

Ex cathedra; including exercises in class, practice with pen and paper or with a computer, and a project

Expected student activities

During the semester, the students are expected to:
• attend the lectures in order to ask questions and interact with the professor,
• attend the exercises session to solve and discuss exercises abou the recently taught material,
• work on a project during the semester which covers the practical side of building an application using a database system,
• take a midterm
• take a final exam

Assessment methods

Homework, project, written examinations and continuous control.

Supervision

Office hours   Yes

| Assistants | Yes |
| Forum | Yes |

Resources

Bibliography

Slides, list of books, additional material (research articles), all indicated and/or available on moodle page.

Ressources en bibliothèque

• Database Management Systems / Ramakrishnan

Notes/Handbook

The slides that are used in the class are available for the students.

Websites

• http://dias.epfl.ch/courses
• http://moodle.epfl.ch/

Moodle Link

• http://moodle.epfl.ch/course/view.php?id=198

CS-307   **Introduction to multiprocessor architecture**
                Falsafi Babak

Cursus	Sem.	Type
Informatique	BA5	Obl.
Systèmes de communication	BA5	Opt.

Language	English
Credits	3
Session	Winter
Semester	Fall
Exam	During the semester
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	1 weekly

Summary

Multiprocessors are a core component in all types of computing infrastructure, from phones to datacenters. This course will build on the prerequisites of processor design and concurrency to introduce the essential technologies required to combine multiple processing elements into a single computer.

Content

- Forms of parallelism
- Parallel programming models
- Cache coherence
- Memory consistency
- Synchronization
- Interconnection networks
- Software efficiency & optimization
- GPU architecture & programming

Keywords

Multiprocessors, multicores, manycores, cache coherence, memory consistency models, memory ordering, manycore cache hierarchies, interconnection networks, synchronization, parallelism, GPU

Learning Prerequisites

Required courses

CS-206 Concurrency
CS-208 Computer architecture / Architecture des Ordinateurs

Important concepts to start the course

Introductory understanding of computer architecture & organization
Basic C/C++ systems programming

Learning Outcomes

By the end of the course, the student must be able to:

- Detect and address inefficiencies in parallel software
- Design and evaluate software for multiple parallel platforms
- Design and evaluate hardware for shared memory
- Compare and contrast hardware design choices in parallel platforms
- Demonstrate and describe the operation of snooping and directory coherence protocols

Teaching methods

Lectures, homework and project

Assessment methods

mid-term and final

Supervision

Office hours   Yes
Assistants   Yes

Resources

Websites

• http://parsa.epfl.ch/courses/cs307/

CS-323 **Introduction to operating systems**

Zwaenepoel Willy

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA6	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	During the semester
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Introduction to basic concepts of operating systems.

Content

Function and general structure of an operating system.
Process management.
Memory management.
File systems.
Virtualization and virtual machines.

Keywords

Operating systems

Learning Prerequisites

Required courses

CS-206 Parallelisme and concurrency
CS-207 Systems programming

Learning Outcomes

By the end of the course, the student must be able to:
 • Manage key components of operating system

Teaching methods

Lectures and exercises.

Expected student activities

Attendance at lectures and completing exercises.

Assessment methods

Midterm and final during the semester.
50% midterm, 50% final.

Supervision

Office hours   Yes

Assistants	Yes
Forum	Yes

Resources

Bibliography

Operating Systems : Three Easy Pieces, R. Arpaci-Dusseau and A. Arpaci-Dusseau (free online book).
Slides available on Moodle.

Ressources en bibliothèque

- Operating Systems : Three Easy Pieces / Arpaci-Dusseau

Références suggérées par la bibliothèque

- Operating Systems Concepts / Silberschatz

COM-300   **Modèles stochastiques pour les communications**
Thiran Patrick

Cursus	Sem.	Type
Bioingénierie	MA1, MA3	Opt.
HES -SC	H	Obl.
Informatique	BA5	Opt.
Mineur en Systèmes de communication	H	Opt.
Sciences du vivant	MA1, MA3	Opt.
Systèmes de communication	BA5	Obl.

Langue	français
Crédits	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

L'objectif de ce cours est la maitrise des outils des processus stochastiques utiles pour un ingénieur travaillant dans les domaines des systèmes de communication, de l'analyse de données et de l'informatique.

Contenu

- Rappels de probabilité: axiomes de probabilité, variable aléatoire et vecteur aléatoire.

- Quelques inégalités utiles.

- Processus stochastiques à temps continu et à temps discret : analyse du second ordre (stationarité, ergodisme, densité spectrale, relations de Wiener-Khintchine, réponse d'un système linéaire invariant à des entrées aléatoires, processus gaussien, processus ARMA, filtres de Wiener). Exemples d'application à des cas simples de détection optimale ou de traitement d'image.

- Processus de Poisson et bruit impulsif de Poisson. Exemple d'application aux transmissions sur fibres optiques.

- Chaînes de Markov à temps discret. Chaînes ergodiques, comportement asymptotique, chaînes absorbantes, temps d'atteinte, marches aléatoires simples, processus de branchement. Exemple d'application à l'analyse d'un algorithme ou système informatique.

- Chaînes de Markov à temps continu. Processus de naissance et de mort à l'état transitoire et stationnaire. Files d'attente simples: définition, loi de Little, files M/M/1... M/M/s/K, M/G/1.

Mots-clés

Probabilité, Processus stochastique, Moments, stationarité, Processus gaussien, Processus de Poisson, Chaîne de Markov, File d'attente.

Compétences requises

Cours prérequis obligatoires

- Algèbre linéaire (MATH 111 ou équivalent).
- Analyse I, II, III (MATH 101, 106, 203 ou équivalent).
- Premier cours de probabilité (MATH 232 ou équivalent)
- Circuits et systèmes II (EE 205 ou équivalent), ou Signaux et systèmes (MICRO310/311 ou équivalent), pour les notions de base de théorie des systèmes linéaires.

Cours prérequis indicatifs

- Circuits et systèmes I (EE 204 ou équivalent) pour les notions de base de théorie des circuits.
- Analyse IV (MATH 204 ou équivalent) pour les notions d'analyse complexe.

Concepts importants à maîtriser

Notions d'algèbre linéaire, en particulier opérations matricielles (inversion, diagonalisation, vaelurs propres d'une matrice).
Notions d'analyse (fonctions d'une ou plusieurs variables réelles, suites et séries, équations différentielles ordinaires linéaires).

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Modéliser un système aléatoire.
- Analyser un problème avec une composante aléatoire.
- Evaluer les solutions d'un problème avec une composante aléatoire.

Méthode d'enseignement

- Ex cathedra (au tableau), 4h par semaine.
- Séances d'exercices, 2h par semaine.

Travail attendu

- Exercices en séance et à domicile

Méthode d'évaluation

- Examens intermédiaires 20%
- Examen final 80%

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

Polycopié; textes de référence sur la page web du cours.

Polycopiés

Polycopié disponible au début du cours.

Sites web

- http://moodle.epfl.ch/course/view.php?id=14236

Préparation pour

Tous les cours en systèmes de communication et informatique (Bachelor et Master) utilisant des modèles stochastiques ou des méthodes aléatoires.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CS-328 **Numerical methods for visual computing**

Jakob Wenzel Alban, Rhodin Helge Jochen

Cursus	Sem.	Type
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Visual computing disciplines are characterized by their reliance on numerical algorithms to process large amounts of visual information such as geometry, images, and volume data. This course will familiarize students with a range of essential numerical tools to solve practical problems in this area

Content

This course provides a first introduction to the field of numerical analysis with a strong focus on visual computing applications. Using examples from computer graphics, geometry processing, computer vision, and computational photography, students will gain hands-on experience with a range of essential numerical algorithms.
The course will begin with a review of important considerations regarding floating point arithmetic and error propagation in numerical computations. Following this, students will study and experiment with several techniques that solve systems of linear and non-linear equations. Since many interesting problems cannot be solved exactly, numerical optimization techniques constitute the second major topic of this course. Students will learn how principal component analysis can be leveraged to compress or reduce the dimension of large datasets to make them easier to store and analyze. The course concludes with a review of numerical methods that make judicious use of randomness to solve problems that would otherwise be intractable.
Students will have the opportunity to gain practical experience with the discussed methods using programming assignments based on Scientific Python.

Keywords

Visual computing, numerical linear algebra, numerical analysis, optimization, scientific computing

Learning Prerequisites

Required courses

MATH-101 (Analysis I) and MATH-111 (Linear Algebra).

Recommended courses

The courses CS-211 (Introduction to visual computing) and MATH-106 (Analysis II) are recommended but not required.

Important concepts to start the course

Students are expected to have good familiarity with at least one programming language (e.g. C/C++, Java, Scala, Python, R, Ruby...). The course itself will rely on Python, but this is straightforward to learn while taking the course.
During the first weeks of the semester, there will be tutorial sessions on using Python and Scientific Python.

Learning Outcomes

By the end of the course, the student must be able to:

- Develop computer programs that use numerical linear algebra and analysis techniques to transform and visualize data.

- Reason about ways of structuring numerical computations efficiently.

- Analyze the numerical stability of programs built on top of floating point arithmetic.

- Recognize numerical problems in visual computing applications and cast them into a form that can be solved or optimized.

Teaching methods

Lectures, interactive demos, theory and programming exercises

Expected student activities

Students are expected to study the provided reading material and actively participate in class and in exercise sessions. They will be given both theoretical exercises and a set of hands-on programming assignments.

Assessment methods

1. Continuous assessment during the semester via project assignments (50%)
2. Final exam (50%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Slides and other resource will be provided in class.
The course textbook is
Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics by Justin Solomon (freely available at the following link: http://people.csail.mit.edu/jsolomon/share/book/numerical_book.pdf)
An optional reference is
Scientific Computing: An Introductory Survey (2nd edition) by Michael Heath

Ressources en bibliothèque

- Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics / Solomon

- Scientific Computing: An Introductory Survey / Heath

Websites

- https://rgl.epfl.ch/courses/NMVC16

Prerequisite for

Although it is not a strict prerequisite, this course is highly recommended for students who wish to pursue studies in the area of Visual Computing, in particular: CS-341 (Introduction to computer graphics), CS-440 (Advanced computer graphics), CS-442 (Computer vision), CS-413 (Computational Photography), CS-444 (Virtual Reality), and CS-445 (Digital 3D geometry processing)

CS-323(a)   **Operating systems implementation**
                   Zwaenepoel Willy

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**2 weekly**
Project	2 weekly

Summary

Implementation of basic concepts of operating systems in Linux

Content

Implementation of system calls, interrupt handling, process and memory management and file systems

Keywords

Operating systems implementation, Linux

Learning Prerequisites

Required courses

CS-206 Parallelisme and concurrency
CS-207 Systems programming
CS-323 Operating systems (preferably to be taken concurrently)

Strictly no admission without concurrent or prior CS-323.

Learning Outcomes

By the end of the course, the student must be able to:
 • Implement key components of operating systems

Teaching methods

Introduction and discussion of assignments in exercise sessions.

Expected student activities

Attendance at exercise sessions.
Implementation of serveral programming projects.

Assessment methods

Programming assignments.
Equal value for each assignment.

Supervision

Office hours	Yes
Assistants	Yes

Forum   Yes

Resources

Bibliography

Operating Systems : Three Easy Pieces, R. Arpaci-Dusseau and A. Arpaci-Dusseau (free online book).
Linux Kernel Programming, 3rd Edition, R. Love.
Slides available on Moodle.

Ressources en bibliothèque

- Linux Kernel Programming / Love
- Operating Systems : Three Easy Pieces / Arpaci-Dusseau

CS-206

Parallelism and concurrency

Kuncak Viktor, Odersky Martin

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA4	Obl.
Systèmes de communication	BA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	1 weekly
Exercises	1 weekly
Project	2 weekly

Summary

The course introduces parallel programming models, algorithms, and data structures, map-reduce frameworks and their use for data analysis, as well as shared-memory concurrency.

Content

Parallel programming & execution models
Functional parallelism
Data-level parallelism
Threads and fork/join parallelism
Synchronization
Cache coherence
Memory models
Threads and Shared Memory in Java
Performance optimization

Keywords

Parallelism, threads, synchronization, locks, memory models.

Learning Prerequisites

Required courses

• Functional programming (CS-210)
• Algorithms (CS-250)
• Computer Architecture (CS-208)

Recommended courses

System oriented programming (CS-207)

Important concepts to start the course

Functional programming and functional data structures
Algorithms and data structures

Learning Outcomes

By the end of the course, the student must be able to:

• Construct parallel software.
• Perform tuning parallel software.

Teaching methods

Ex cathedra, labs, exercices

Assessment methods

With continuous control

Resources

Notes/Handbook

Lecture notes, copies of the slides

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

COM-302 **Principles of digital communications**

Telatar Emre

Cursus	Sem.	Type
HES -SC	E	Obl.
Informatique	BA6	Opt.
Mineur en Systèmes de communication	E	Opt.
Systèmes de communication	BA6	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

This course is on the foundations of digital communication. The focus is on the transmission problem (rather than being on source coding).

Content

Optimal receiver for vector channels
Optimal receiver for waveform (AWGN) channels
Various signaling schemes and their performance
Efficient signaling via finite-state machines
Efficient decoding via Viterbi algorithm
Communicating over bandlimited AWGN channels
Nyquist Criterion
Communicating over passband AWGN channels

Keywords

Detection, estimation, hypothesis testing, Nyquist, bandwidth, error probability, coding, decoding, baseband, passband, AM, QAM, PSK.

Learning Prerequisites

Required courses

Signal processing for communications and modèles stochastiques pour les communications

Important concepts to start the course

Linear algebra, probability.

Learning Outcomes

By the end of the course, the student must be able to:

- Estimate the error probability of a communication link
- Design a "physical layer" communication link
- Implement a prototype of a "physical layer" transmitter/receiver via Matlab

Teaching methods

Ex cathedra + exercises + project. Lots of reading at home and exercises in class.

Assessment methods

With continuous control

Resources

Websites

• http://moodle.epfl.ch

Prerequisite for

Advanced Digital Communications
Software-Defined Radio: A Hands-On Course

MATH-232

Probabilités et statistique

Thibaud Emeric Rolland Georges

Cursus	Sem.	Type
HES - IN	E	Obl.
HES -SC	E	Obl.
Informatique	BA4	Obl.
Systèmes de communication	BA4	Obl.

Langue	français
Crédits	6
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo

Résumé

Un cours d'introduction aux probabilités et à la statistique.

Contenu

- Eléments de combinatoire : notions de théorie des ensembles et dénombrement.
- Probabilités élémentaires : expérience aléatoire, espace de probabilité, probabilité conditionnelle, indépendance.
- Variables aléatoires : densité et fonction de masse, lois de probabilités usuelles, moyenne, variance, corrélation et covariance, fonction génératrice des moments, distribution conjointe, distribution conditionnelle et marginale, transformations.
- Limites de variables aléatoires : notions de convergence, loi des grands nombres, théorème central limite.
- Statistique descriptive.
- Inférence statistique : estimation, intervalles de confiance, tests d'hypothèses, vraisemblance, modélisation statistique, inférence Bayésienne.

Compétences requises

Cours prérequis obligatoires

- Analyse I, II
- Algèbre linéaire

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Calculer des probabilités simples en utilisant des arguments de combinatoire.
- Dériver es propriétés d'une variable aléatoire.
- Proposer des méthodes appropriées pour décrire et visualiser des données.
- Définir des modèles de probabilités appropriés pour des problèmes simples.
- Estimer les paramètres d'un modèle et construire des intervalles de confiance.

Méthode d'enseignement

Cours ex-cathedra, exercices en salle

Méthode d'évaluation

Quizzes, test intermédiaire, examen final

Encadrement

Assistants   Oui

Ressources

Service de cours virtuels (VDI)
Non

Bibliographie
Ross, S. (2012) A first course in probability (9th edition). Pearson.
Aussi disponible en traduction française (PPUR): "Initiation aux probabilités".

Ressources en bibliothèque

- Initiation aux probabilités / Ross
- A first course in probability / Ross

Polycopiés
Un polycopié des notes de cours avec exercices sera disponible au début du semestre.

CS-207   **Programmation orientée système**

Chappelier Jean-Cédric

Cursus	Sem.	Type
HES - IN	E	Obl.
HES -SC	E	Obl.
Informatique	BA4	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA4	Obl.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	1 hebdo
Exercices	2 hebdo

Résumé

Cours de programmation en langage C se focalisant sur l'utilisation des ressources système, en particulier la gestion de la mémoire (pointeurs).

Contenu

Initiation à la programmation en C : variables, expressions, structures de contrôle, fonctions, entrées-sorties, ...

Approfondissement des spécificités de la programmation système rudimentaire : gestion de la mémoire (pointeurs), des fichiers et autres entrées/sorties.

Les concepts théoriques introduits lors des cours magistraux seront mis en pratique dans le cadre d'exercices sur machine.

Mots-clés

Programmation, langage C, pointeurs, gestion mémoire

Compétences requises

Cours prérequis indicatifs

Introduction à la programmation (CS-107) + Pratique de la programmation orientée-objet (CS-108)

Concepts importants à maîtriser

bases de programmation

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Modéliser en langage C une situation simple du monde réelle décrite en Français
- Analyser un problème complexe relatifs aux systèems d'information pour le décomposer en sous-problèmes
- Concevoir des algorithmes résolvant des tâches simple à avancées relatives au systèmes informatiques
- Réaliser de façon autonome une application de petite taille au moyen du langage C
- Analyser du code C pour en décrire le résultat ou le corriger s'il est erroné
- Tester l'adéquation du résultat d'un programme par rapport à la tâche demandée
- Transcrire un algorithme en son programme équivalent en C

Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.
- Accéder aux sources d'informations appropriées et les évaluer.
- Utiliser une méthodologie de travail appropriée, organiser un/son travail.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Gérer ses priorités.

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateur

Travail attendu

participation au cours et aux exercices ; travail personnel à la maison.

Méthode d'évaluation

1 exercice à rendre (10 %), 1 série pratique notée (40%) et examen final papier (50%)

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)
Oui

Bibliographie
Notes de cours

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=6731

Préparation pour

Introduction au bases de données (CS-322) ; Concurrence (CS-206) ; Systèmes d'exploitation (CS-323) ; Genie logiciel (CS-305)

CS-309 **Projet de systems-on-chip**

Beuchat René

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	90h
Semaines	14
Heures	**3 hebdo**
Projet	3 hebdo

Résumé

L'objectif de ce cours est de s'approprier les connaissances nécessaires pour réaliser du développement "full stack" depuis le hardware jusqu'au software application et s'exécutant sur un système d'exploitation Linux. La base des laboratoires est une plateforme embarquée basée sur un SOC FPGA & ARM.

Contenu

Ce cours-labos va surtout permettre aux étudiants de comprendre les liens entre une application, un operating system (Linux) et le matériel informatique. Une plateform ayant une FPGA et 2 processeurs ARM intégrés est utilisée comme support des labos.
Ce cours est centré sur des laboratoires pratiques à réaliser par les étudiants. Généralement une introduction d'une heure est suivie de 1 à 3 sessions pour réaliser des mini-projets par groupes.
Lors des laboratoires, les travaux seront effectués sur des cartes FPGA-SOC avec processeurs embarqués sous forme softcore (NIOSII) et/ou hardcore (ARM). Des interfaces programmables simples et complexes seront développées en VHDL pour s'interfacer avec des modules externes à contrôler. La méthodologie pas-à-pas sera utilisée pour arriver à la réalisation de systèmes relativement complexes.
Les sujets suivants seront étudiés et au final une application complète sera implémentée :
1. Analyse du système : Multicore ARMs, FPGA, I/Os, et interfaces programmables spécialisées ;
2. Design et simulation d'une interface programmable en VHDL, pour un ADC permettant de lire des joysticks analogiques, et un contrôle de servo-moteurs ;
3. Design et simulation d'un accélérateur spécifique en VHDL pour une caméra thermique ;
4. Test du système spécifique avec développement de logiciel en C avec des outils de cross-debugging (baremetal coding) ;
5. Boot et test d'un système embarqué, baremetal design ;
6. Installation d'un OS: Adaptation et compilation de Linux pour la carte de laboratoire ;
7. Mini-projet final où les étudiants utiliseront leurs connaissances pour capturer une image thermique et à l'aide de joystick définiront divers paramètres et afficheront les résultats sur un affichage LCD ou écran VGA.

Mots-clés

Micro-controllers, Micro-contrôleurs, FPGA, SOC, Embedded Systems, Logic Analyzer, Oscilloscope, projects, work in groups, C, VHDL.

Compétences requises

Cours prérequis obligatoires

- CS-173: Conception de systèmes numériques
- CS-208: Architecture des ordinateurs
- CS-209: Architecture des Systems-on-Chip
- CS-307: Introduction to multiprocessor architecture

Concepts importants à maîtriser

Les étudiants doivent connaître l'architecture d'un processeur, d'un système informatique, quelques notions de programmation en C, en langage assembleur et en VHDL.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Analyser le problème à résoudre ;
- Planifier le travail à réaliser ;
- Créer l'architecture du système à réaliser ;
- Coordonner le travail ;
- Concevoir le système à réaliser ;
- Implémenter la solution ;
- Intégrer les modules disponibles ;
- Expliquer la solution développée.

Compétences transversales

- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Fixer des objectifs et concevoir un plan d'action pour les atteindre.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.
- Faire preuve d'esprit critique
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.

Méthode d'enseignement

- Travail par groupes d'étudiants ;
- Laboratoires pratiques ;
- Mini-projets spécifiques par groupe ;
- Travaux avec des outils de développement croisé et de systèmes matériels réels ;
- Utilisation des outils de débogage tels que oscilloscopes et analyseurs logiques ;
- Les travaux sont réalisés sur les systèmes matériels réels avec des micro-contrôleurs et FPGA.

Travail attendu

- Brainstorming par groupe ;
- Répartition de la charge de travail à travers le groupe ;
- Gestion de la planification ;
- Analyse des données des data-sheet des composants utilisés ;
- Résolution de problèmes pratiques sur des systèmes matériels ;
- Développement de modules FPGA en VHDL ;
- Développement de logiciels en C et observation en assembleur du déroulement du programme ;
- Configuration et construction d'un Linux sur un système embarqué.

Méthode d'évaluation

1. Rapports réguliers (70%)
2. Présentation orale finale (15%)

3. Démonstration (15%)

Encadrement

Office hours   Non
Assistants   Oui
Forum électronique   Oui

Ressources

Service de cours virtuels (VDI)

Non

Sites web

• http://wiki.epfl.ch/prsoc

Liens Moodle

• http://moodle.epfl.ch/course/view.php?id=14480

Préparation pour

CS-473: Embedded Systems
CS-476: Real Time Embedded systems

CS-398

Projet en informatique I

Profs divers *

Cursus	Sem.	Type
Informatique	BA5, BA6	Obl.

Langue	français
Crédits	8
Session	Hiver, Eté
Semestre	Automne
Examen	Pendant le semestre
Charge	240h
Semaines	14
Heures	**2 hebdo**
Projet	2 hebdo

Résumé

Travaux de recherche individuelle à effectuer pendant le semestre, selon les directives d'un professeur ou d'un assistant.

Contenu

Sujet de travail à choisir parmi les domaines proposés sur le site web :

http://ic.epfl.ch/page-20200-fr.html

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Organiser un projet
- Evaluer sa progression au cours du projet
- Présenter un projet

Compétences transversales

- Ecrire un rapport scientifique ou technique.
- Ecrire une revue de la littérature qui établit l'état de l'art.

Méthode d'évaluation

Rapport écrit et présentation orale

Encadrement

Office hours	Non
Assistants	Non
Forum électronique	Non

Ressources

Service de cours virtuels (VDI)
Non

Sites web

- http://ic.epfl.ch/page-17123-fr.html

CS-207(a)   **Projet programmation système**

Chappelier Jean-Cédric, Figueiredo Mascarenhas Moreira Pedrosa Luis David

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA4	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA4	Opt.

Langue	français
Crédits	2
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	60h
Semaines	14
Heures	**2 hebdo**
Projet	2 hebdo

Résumé

L'objectif de ce cours à projet est de donner aux étudiants une expérience de la pratique de la programmation système : écriture, correction, amélioration et analyse critique de leur code.

Contenu

- Ce cours sera enseigné en parallèle du cours « Programmation Orientée Système » (CS-207). Il offre aux étudiants l'opportunité de développer dans un cadre pratique à large échelle les concepts présentés dans cet autre cours. Les étudiants devront en effet non seulement développer leur propre code à partir de rien sur un cas concret, mais aussi lire du code professionnel développé par d'autres de sorte à pouvoir s'en inspirer.

- Ce cours consistera en un projet constitué de plusieurs parties réparties sur le semestre. Il insistera sur les concepts clés présentés dans les autres cours liés au domaine de la programmation système (systèmes de fichiers, réseaux, accès mémoire, concurrence, …), mais mettra aussi en place de façon pratique plusieurs éléméns qui seront approfondis plus tard dans le cursus (aspects réseaux avancés, planification, etc.) dans le but d'offrir aux étudiants une première approche pratique à ces concepts.

Mots-clés

- programmation système, gestion mémoire, système de fichiers

Compétences requises

Cours prérequis obligatoires

- Programmation Orientée Système (CS-207)
- Parallelism & Concurrency (CS-206)

Cours prérequis indicatifs

- Computer Networks

Concepts importants à maîtriser

-
programmation
-
parallélisme et concurrence

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Modéliser en langage C des problèmes système typiques
- Analyser des problèmes complexes relatifs aux sytèmes informatiques et les décomposer en sous-problèmes
- Réaliser des applications de taille moyenne en langage C
- Analyser des projet en langage C pour être capable de comprendre ce qu'ils font et comment ils sont organisés
- Tester les résultat d'un projet en langage C et le corriger là où cela est nécessaire

Compétences transversales

- Accéder aux sources d'informations appropriées et les évaluer.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Gérer ses priorités.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.

Méthode d'enseignement

projet

Travail attendu

-
écrire le code et la documentation d'un projet de groupe

Méthode d'évaluation

-
rendu du projet en plusieurs étapes progressives au cours du semestre dont 40 % pour le rendu principal.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Préparation pour

-
CS-323 (Operating Systems)

COM-301 **Sécurité des réseaux**

Oechslin Philippe

Cursus	Sem.	Type
Information security minor	H	Opt.
Informatique	BA5	Obl.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA5	Obl.
UNIL - Sciences forensiques	H	Obl.

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo

Résumé

Comprendre les menaces présentes dans les réseaux informatiques et savoir comment protéger un réseau par des moyens techniques et organisationnels.

Contenu

Menaces :
- Spam, phishing, virus, chevaux de Troie, dénis de service,exploitation de failles.

Mesures de protection :
- Firewalls, proxys, anti-virus, détection d'intrusion

Protocoles et applications :
- Messageries sécurisés (PGP, S/MIME)
- IPSec, HTTPS, SSL/TLS, SSH, WPA

Aspects organisationnels :
- Analyse de risques et politique de sécurité
- Normes et standards

Aspects réglementaire :
- Droit concernant les systèmes d'information

Compétences requises

Cours prérequis indicatifs

Notions de base de TCP/IP
Notions de base de programmation

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Discuter les mécanismes des attaques informatiques.
- Evaluer l'impact d'une attaque informatique.
- Concevoir des mesures de protection d'un réseau informatique.
- Expliquer les protocoles cryptographiques vu au cours.

- Analyser un protocole cryptographique basé sur des primitives standard.
- Décrire un système de gestion de la sécurité de l'information.

Méthode d'enseignement

Ex cathedra et exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Forum électronique   Oui

Ressources

Bibliographie

Sécurité Informatique, cours et exercices corigés, Gildas Avoine, Pascal Junod et Philippe Oechslin, 3ème édition, Vuibert, 2015 , ISBN : 978-2-311-40168-4

Ressources en bibliothèque

- Sécurité Informatique / Avoine, Junod and Oechslin

Sites web

- http://lasec.epfl.ch/teaching.shtml

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=14728

COM-303   **Signal processing for communications**

Prandoni Paolo

Cursus	Sem.	Type
Auditeurs en ligne	E	Obl.
HES -SC	E	Obl.
Informatique	BA6	Opt.
Mineur en Systèmes de communication	E	Opt.
Science et ing. computationelles	MA2	Opt.
Systèmes de communication	BA6	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

Students learn digital signal processing theory, including discrete time, Fourier analysis, filter design, sampling, interpolation and quantization; they are introduced to image processing and data communication system design.

Content

1. Basic discrete-time signals and systems: signal classes and operations on discrete-time signals, signals as vectors in Hilbert space
2. Fourier Analysis: properties of Fourier transforms, DFT, DTFT; FFT.
3. Discrete-Time Systems: LTI filters, convolution and modulation; difference equations; FIR vs IIR, stability issues.
4. Z-transform: properties and regions of convergence, applications to linear systems.
5. Filter Design: FIR design methods, IIR design methods, filter structures.
6. Stochastic Signal Processing: random processes, spectral representation.
7. Interpolation and Sampling: the continuous-time paradigm, interpolationthe sampling theorem, aliasing.
8. Quantization: A/D and D/A converters.
9. Multi-rate signal processing: upsampling and downsampling, oversampling.
10. Multi-dimensional signals and processing: introduction to Image Processing.
11. Practical applications: digital communication system design, ADSL.

Keywords

signal processing, discrete-time, continuous-time, filter, filter design, sampling, aliasing, DSP, Fourier transform, FFT, modem, ADSL

Learning Prerequisites

Required courses

calculus, linear algebra

Recommended courses

Circuits and systems, basic probability theory

Important concepts to start the course

vectors and vector spaces, functions and sequences, infinite series

Learning Outcomes

By the end of the course, the student must be able to:
- Identify signals and signal types
- Recognize signal processing problems

- Apply the correct analysis tools to specific signals
- Check system stability
- Manipulate rational transfer functions
- Implement signal processing algorithms
- Design digital filters
- Interpret complex signal processing systems

Transversal skills

- Use a work methodology appropriate to the task.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Use both general and domain specific IT resources and tools

Teaching methods

Course with exercises in class and on the computer

Expected student activities

complete weekly homework, write numerical routines to implement core concepts

Assessment methods

midterm exam for bonus points and final exam for final grade.

Resources

Bibliography

Signal processing for Communications, EPFL Press, 2008, by P. Prandoni and M. Vetterli. The book is available for sale in printed form online and in bookstores; in iBook format on the Apple store and is also available as a free pdf file at http://www.sp4comm.org/

Ressources en bibliothèque

- Signal processing for Communications / Prandoni

Websites

- http://lcav.epfl.ch/sp4comm
- http://www.sp4comm.org/

Prerequisite for

adaptive signal processing, image processing, audio processing, advanced signal processing

CS-305 **Software engineering**

Candea George

Cursus	Sem.	Type
HES - IN	H	Obl.
Informatique	BA5	Obl.
Mineur en Informatique	H	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Systèmes de communication	BA5	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Project	3 weekly

Summary

Covers basic aspects of modern software development practices and tools, foundations of software engineering, how to think about software, structure it, modify it, and improve it. Covers the software development process (including agile methods) and working as part of a team of developers.

Content

- Object-oriented design and reasoning
- Design patterns
- Principles of building reliable and secure software
- Performance tuning
- Testing and debugging
- Code layout and style
- Development processes
- Software project management
- Tools for writing, analyzing, and debugging code, as well as source code management

Being a good software engineer entails a continuous learning process. Unlike mathematics or physics, this field changes fast, thus making continuous and independent learning essential. This course prepares students to become lifelong auto-didacts that build upon the foundation of imutable principles governing good software engineering.

Keywords

software development, software engineering, software design, software development tools, development processes, agile methods

Learning Prerequisites

Required courses

First and foremost, students taking Software Engineering must be proficient Java programmers. Without a good prior knowledge of Java, it is very difficult to keep up with the pace of the class. Furthermore, this course builds on material taught in the courses below. Students who do not master the material taught in the prerequisite courses prior to starting Software Engineering typically do not manage to pass:

- CS-108 Practice of Object-Oriented Programming
- CS-210 Functional Programming
- CS-206 Parallelism and concurrency
- CS-207 System-oriented Programming

Recommended courses

The material in the following courses is helpful but not absolutely required:

• COM-208 Computer networks
• CS-208/209 Computer architecture

Important concepts to start the course

• Good knowledge of object-oriented programming (e.g., in Java)
• Knowledge of using version control systems (e.g., Git)
• Using modern development tools (e.g., IDE, Android emulator)

Learning Outcomes

By the end of the course, the student must be able to:

• Design software that is reliable, secure, user-friendly, and performs well
• Implement (in software) sophisticated designs and algorithms
• Specify requirements for software systems
• Develop code that is maintainable
• Organize a team of developers to execute a medium-sized software project
• Assess / Evaluate design and implementation options
• Choose alternatives to optimize for an objective (e.g., performance)

Transversal skills

• Plan and carry out activities in a way which makes optimal use of available time and other resources.
• Set objectives and design an action plan to reach those objectives.
• Assess progress against the plan, and adapt the plan as appropriate.
• Manage priorities.
• Evaluate one's own performance in the team, receive and respond appropriately to feedback.
• Give feedback (critique) in an appropriate fashion.
• Resolve conflicts in ways that are productive for the task and the people concerned.
• Assess one's own level of skill acquisition, and plan their on-going learning goals.
• Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.

Teaching methods

• Combination of online and in-class lectures
• Recitations and lab sessions
• Homework exercises
• Course project

Expected student activities

• Watch online lectures
• Attend and participate in lectures and recitations

- Read and understand assigned materials
- Complete programming assignments and attend lab sessions
- Work with team members to complete a substantial project

Assessment methods

Throughout the semester (contrôle continu). The final grade will be determined approximately as follows:

- 10% for 2 homework assignments done individually
- 50% for 1 team project
- 40% based on 2 exams during the semester and potentially online quizzes

Exact formula may vary from year to year.

Supervision

Office hours   Yes
Assistants   Yes
Forum   Yes
Others   See http://sweng.epfl.ch/

Resources

Virtual desktop infrastructure (VDI)
No

Bibliography
See http://sweng.epfl.ch for up-to-date bibliography

Ressources en bibliothèque

- Code Complete: A Practical Handbook of Software Construction / McCornell
- The Deadline: A Novel About Project Management / DeMarco

Websites

- http://sweng.epfl.ch/

CS-251   **Theory of computation**
                Vishnoi Nisheeth

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA4	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA4	Obl.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

This course constitutes an introduction to theory of computation. It discusses the basic theoretical models of computing (finite automata, Turing machine), as well as, provides a solid and mathematically precise understanding of their fundamental capabilities and limitations.

Content

- Basic models of computation (finite automata, Turing machine)
- Elements of computability theory (undecidability, reducibility)
- Introduction to time complexity theory (P, NP and theory of NP-completeness)

Keywords

theory of computation, Turing machines, P vs. NP problem, complexity theory, computability theory, finite automata, NP-completeness

Learning Prerequisites

Required courses

CS-101 Advanced information, computation, communication I
CS-250 Algorithms

Learning Outcomes

By the end of the course, the student must be able to:

- Perform a rigorous study of performance of an algorithm or a protocol
- Classify computational difficulty of a decision problem
- Define the notion of NP-completeness
- Analyze various computation models
- Design a reduction between two computational problems
- Characterize different complexity classes
- Explain P vs. NP problem

Transversal skills

- Use a work methodology appropriate to the task.

• Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Ex cathedra with exercises

Assessment methods

Written exam and continuous control

COM-309 | **Traitement quantique de l'information**

Macris Nicolas

Cursus	Sem.	Type
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	3 hebdo
Exercices	1 hebdo

Résumé

L'information est traitée et stockée dans des composants matériels. Avec leur miniaturisation, il faut remplacer le concept de bit classique doit être remplacé par la notion de bit quantique. Ce cours développe le sujet des communications, de la cryptographie et des corrélations quantiques.

Contenu

Introduction a la mecanique quantique des systemes discrets.
- Polarisation des photons, états quantiques, règle de Born.
- Expérience de Stern-Gerlach, spin 1/2, états quantiques sur la sphère de Bloch.
- Dynamique du spin, Oscillations de Rabi et manipulations de l'état quantique.
- Notion abstraite de qubit. Etats à plusieurs qubits.
Cryptographie, Communications et Corrélations
- Génération d'une clé secrète: protocoles BB84 et B92.
- Intrication: paires de Einstein-Podolsky-Rosen.
- Inégalités de Bell. Expériences d'Aspect-Grangier. Protocole de Ekert pour une clé secrète.
- Protocoles de téléportation et dense coding.

Mots-clés

Polarisation, spin, bit quantique, intrication, téléportation, cryptographie quantique.

Compétences requises

Cours prérequis obligatoires
Algèbre linéaire.

Cours prérequis indicatifs
physique de base: mécanique et ondes.

Concepts importants à maîtriser
Matrices, valeurs et vecteurs propres, produit scalaire, manipulations algébriques de base avec des nombres complexes.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Expliquer les principes de la physique quantique pour les systèmes discrets
- Expliquer le concept de qubit et donner quelques exemples
- Décrire comment manipuler des qubits

- Connaitre les protocoles de base de la cryptographie quantique.
- Connaitre les protocoles de dense coding et de téléportation.
- Expliquer ce qu'est l'intrication.

Méthode d'enseignement

Ex-Cathedra. Séances d'exercices. Discussions des lectures pédagogiques proposées aux étudiants.

Travail attendu

Participation au cours. Résolution d'exercices. Lectures de revues pédagogiques.

Méthode d'évaluation

midterm 20%, 4 graded homeworks 20%, examen final écrit 60%.

Ressources

Bibliographie

David Mermin, ***Quantum computer science, An introduction***, *Cambridge university press 2000.* Livre écrit pour des informaticiens et qui ne requiert pas de connaissances en physique..
Michel Le Bellac, ***A short introduction to quantum information and quantum computation,*** *Cambridge University Press. Pour l'édition en francais voir Editions Belin 2005.* Un livre pédagogique qui introduit quelques aspects physiques elementaires du sujet.
Neil Gershenfeld. ***The physics of information technology.*** *Cambridge University Press.* Un livre original sur les technologies de base utiles en informatique et communication classique et/ou quantique.

Ressources en bibliothèque

- The physics of information technology / Gershenfeld
- A short introduction to quantum information and quantum computation / Le Bellac
- Quantum computer science / Mermin

Polycopiés

Notes fournies en classe. Revues sur le sujet. exercices et corrigés fournis en cours d'année.

Sites web

- https://ipg.epfl.ch/doku.php?id=en:courses

Préparation pour

Calcul quantique

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SECTION D'INFORMATIQUE

Cycle Master

2017 / 2018

CS-450	**Advanced algorithms**

Svensson Ola Nils Anders

Cursus	Sem.	Type
Data Science	MA2	Obl.
Information security minor	E	Opt.
Informatique et communications		Obl.
Informatique	MA2	Obl.
Mineur en Informatique	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Science et ing. computationelles	MA2	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	**7 weekly**
Lecture	4 weekly
Exercises	2 weekly
Project	1 weekly

Summary

A first graduate course in algorithms, this course assumes minimal background, but moves rapidly. The objective is to learn the main techniques of algorithm analysis and design, while building a repertory of basic algorithmic solutions to problems in many domains.

Content

Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, approximation. Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization, linear programming. Examples from graph theory, linear algebra, geometry, operations research, and finance.

Keywords

See content.

Learning Prerequisites

Required courses

An undergraduate course in Discrete Structures / Discrete Mathematics, covering formal notation (sets, propositional logic, quantifiers), proof methods (derivation, contradiction, induction), enumeration of choices and other basic combinatorial techniques, graphs and simple results on graphs (cycles, paths, spanning trees, cliques, coloring, etc.).

Recommended courses

An undergraduate course in Data Structures and Algorithms.
An undergraduate course in Probability and Statistics.

Important concepts to start the course

Basic data structures (arrays, lists, stacks, queues,trees) and algorithms (binary search; sorting; graph connectivity); basic discrete mathematics (proof methods, induction, enumeration and counting, graphs); elementary probability and statistics (random variables, distributions, independence, conditional probabilities); data abstraction.

Learning Outcomes

By the end of the course, the student must be able to:

- Use a suitable analysis method for any given algorithm
- Prove correctness and running-time bounds
- Design new algorithms for variations of problems studied in class

- Select appropriately an algorithmic paradigm for the problem at hand
- Define formally an algorithmic problem

Teaching methods

Ex cathedra lecture, reading

Assessment methods

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	For details, see the course web page.

Resources

Bibliography

See web page for the course.

Ressources en bibliothèque

- Randomized Algorithms / Motwani
- Approximation Algorithms / Vazirani
- Quantum Computation and Quantum Information / Nielsen
- Algebraic Complexity Theory / Buergisser
- Computational Complexity / Papadimitrou

Notes/Handbook

Class notes and references for the running semester will be provided as needed within a few days after each lecture.

Websites

- http://theory.epfl.ch/courses/AdvAlg/

CS-420	**Advanced compiler construction**
	Schinz Michel

Cursus	**Sem.**	**Type**
Informatique	MA2	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

Students learn several implementation techniques for modern functional and object-oriented programming languages. They put some of them into practice by developing key parts of a compiler and run time system for a simple functional programming language.

Content

Part 1: implementation of high-level concepts
- functional languages: closures, continuations, tail call elimination,
- object-oriented languages: object layout, method dispatch, membership test.

Part 2: optimizations
- compiler intermediate representations (RTL, SSA, CPS),
- inlining and simple optimizations,
- register allocation,
- instruction scheduling.

Part 3: run time support
- interpreters and virtual machines,
- memory management (including garbage collection).

Keywords

compilation, programming languages, functional programming languages, object-oriented programming languages, code optimization, register allocation, garbage collection, virtual machines, interpreters, Scala.

Learning Prerequisites

Recommended courses
Compiler Construction

Important concepts to start the course
Excellent knowledge of Scala and C programming languages

Learning Outcomes

By the end of the course, the student must be able to:
- Assess / Evaluate the quality of a compiler intermediate representation
- Design compilers and run time systems for object-oriented and functional programming languages
- Implement rewriting-based compiler optimizations
- Implement efficient virtual machines and interpreters
- Implement mark and sweep or copying garbage collectors

Teaching methods

Ex Cathedra, mini-project

Assessment methods

Continuous control (mini-project 80%, final exam 20%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Ressources en bibliothèque

- Engineering a Compiler / Cooper
- Compiling with continuations / Appel
- The garbage collection handbook : the art of automatic memory management / Jones
- Modern Compiler Implementation in Java / Appel

Websites

- http://cs420.epfl.ch/

CS-470   Advanced computer architecture

Ienne Paolo

Cursus	Sem.	Type
Computer engineering minor	E	Opt.
Génie électrique et électronique	MA2, MA4	Opt.
Information security minor	E	Opt.
Informatique	MA2	Obl.
Mineur en Informatique	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Oral
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

The course studies the most important techniques to exploit Instruction-Level Parallelism and discusses the relation with the critical phases of compilation. It also analyses emerging classes of processors for complex single-chip systems.

Content

Pushing processor performance to its limits:
- Principles of Instruction Level Parallelism (ILP).
- Register renaming techniques.
- Prediction and speculation.
- Simultaneous multithreading.
- VLIW and compiler techniques for ILP.
- Dynamic binary translation.

Embedded processors:
- Specificities over stand-alone processors.
- Overview of DSPs and related compilation challenges.
- Configurable and customisable processors.

Keywords

Processors, Instruction Level Parallelism, Systems-on-Chip, Embedded Systems.

Learning Prerequisites

Required courses

- Architecture des ordinateurs I (coursebook until 2013-2014).
- Architecture des ordinateurs (coursebook since 2014-2015).

Recommended courses

- Architecture des ordinateurs II (coursebook until 2013-2014).
- Architecture des systèmes-on-chip (coursebook since 2014-2015).

Learning Outcomes

By the end of the course, the student must be able to:
- Design strategies to exploit instruction level parallelism in processors.
- Contrast static and dynamic techniques for instruction level parallelism.
- Design effective processor (micro-)architectures for which efficient compilers can be written.

Teaching methods

Courses, labs, and compulsory homeworks.

Assessment methods

Final oral exam.

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Bibliography

- John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufman, 5th edition, 2011.

Ressources en bibliothèque

- Computer Architecture / Hennessy

Websites

- http://lap.epfl.ch/aca
- http://fpga4u.epfl.ch

CS-440 — Advanced computer graphics

Jakob Wenzel Alban

Cursus	Sem.	Type
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
Mineur en Informatique	E	Opt.
Mineur en Systèmes de communication	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

This course covers advanced 3D graphics techniques for realistic image synthesis. Students will learn how light interacts with objects in our world, and how to recreate these phenomena in a computer simulation to create synthetic images that are indistinguishable from photographs.

Content

This is a project-based course: students will initially receive a basic software package that lacks most rendering-related functionality.

Over the course of the semester, we will discuss a variety of concepts and tools including the basic physical quantities, how light interacts with surfaces, and how to solve the resulting mathematical problem numerically to create realistic images. Advanced topics include participating media, material models for sub-surface light transport, and Markov Chain Monte Carlo Methods.

Each major topic is accompanied by an assignment so that students can implement solution algorithms and obtain practical experience with these techniques within their own software framework.

Towards the end of the course, students will realize a self-directed final project that extends their rendering software with additional features of their own choosing. The objective of the final project is to create a single image of both technical and artistic merit that is entered into a rendering competition and judged by an independent panel of computer graphics experts.

Learning Prerequisites

Required courses

Nothing

Recommended courses

Introduction to Computer Graphics

Important concepts to start the course

We will rely on calculus, linear algebra and use basic concepts of algorithms and data structures. Students are expected to be familiar with the C++ programming language that is used in the programming assignments.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize and understand the physical quantities of light transport and be able to perform basic computations using pencil+paper
- Explain a range of surface and subsurface material models

- Explain the rendering and radiative transfer equation and show how to construct Monte Carlo methods to solve them
- Design and implement an advanced rendering system based on Monte Carlo integration
- Assess / Evaluate the performance and conceptual limits of the implemented simulation code

Teaching methods

Lectures, interactive demos, theory and programming exercises, programming project, project tutoring

Expected student activities

The student are expected to study the provided reading material and actively participate in class. They should prepare and resolve the exercises, prepare and carry out the programming project.

Assessment methods

Intermediate assignments (60%), final project (40%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

A list of books will be provided at the beginning of the class

Notes/Handbook

Slides and online resources will be provided in class

Websites

- https://rgl.epfl.ch/courses/ACG17

CS-471   **Advanced multiprocessor architecture**

Falsafi Babak

Cursus	Sem.	Type
Computer engineering minor	H	Opt.
Génie électrique et électronique	MA1, MA3	Opt.
Informatique et communications		Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	4 weekly

Remarque

Cours biennal donné une année sur deux les années impaires (donné en 2017-18)

Summary

Multiprocessors are now the defacto building blocks for all computer systems. This course will build upon the basic concepts offered in Computer Architecture I to cover the architecture and organization of modern multiprocessors from mobile and embedded platforms to servers, data centers and cloud computing platforms.

Content

Introduction to multiprocessor systems, parallel programming models including Pthreads, MPI, hardware and software transactional memory, synchronization primitives, memory consistency mdels, cache coherence, on-chip shared cache architectures, on-chip interconnects, multi-chip interconnects, multi-chip bus-based and general-purpose interconnect-based shared-memory systems, clusters.
The course will include weekly readings, discussions, and student reviews and reports on publications (besides the text book) of seminal and recent contributions to the field of computer architecture. Student reviews, class discussions, and an independent research project will account for a significant fraction of the grade. Feedback on performance will be given only upon request by a student. There will be no recitation classes.
The course will also include an independent and original research project, in which students study, improve, and evaluate multiprocessor innovations using a software simulation infrastructure. There will be a list of project ideas given out, but students can suggest and work on their own ideas with potentials for advancing the state of the art.

Learning Prerequisites

Recommended courses

Computer Architecture I, basic C/C++ systems programming.

Learning Outcomes

By the end of the course, the student must be able to:
- Design and evaluate parallel computer organizations
- Develop parallel programs and benchmarks for parallel systems
- Design the basic components of modern parallel systems including multiple processors, cache hierarchies and networks
- Quantify performance metrics for parallel systems
- Interpret and critique research papers
- Plan , propose and conduct a research project empirically
- Present research contributions

Teaching methods

Lectures, homeworks, and a research project

Assessment methods

Continuous control :
Homework : 30 %, Project 15 %, Midterm test : 20 %,
End term test : 35 %

Resources

Websites

- http://parsa.epfl.ch/courses/cs471/.

COM-417	**Advanced probability and applications**		

Lévêque Olivier

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique et communications		Obl.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly

Summary

In this course, various aspects of probability theory are considered. The first part is devoted to the main theorems in the field (law of large numbers, central limit theorems), while the second part focuses on the theory of martingales in discrete time.

Content

I. Probability
- sigma-fields, probability measures, random variables
- independence, expectation
- convergence of sequences of random variables
- laws of large numbers- central limit theorem
- concentration inequalities
- moments
II. Martingales
- conditional expectation
- definition and properties of a martingale
- stopping times, optional stopping theorem
- maximal inequalities
- convergence theorems

Keywords

probability, measure theory, martingales, convergence theorems

Learning Prerequisites

Required courses

Basic probability course
Calculus courses

Recommended courses

complex analysis

Important concepts to start the course

This course is NOT an introductory course on probability: the students should have a good understanding and practice of basic probability concepts such as: distribution, expectation, variance, independence, conditional probability.
The students should also be at ease with calculus. Complex analyisis is a plus, but is not required.
On the other hand, no prior background on measure theory is needed for this course: we will go through the basic concepts one by one at the beginning.

Learning Outcomes

By the end of the course, the student must be able to:
- Understand the foundations of probability theory
- Acquire a solid knowledge of martingale theory

Teaching methods

Ex cathedra + exercises

Expected student activities

active participation to exercise sessions

Assessment methods

Midterm 10%, homeworks 10%, exam 80%

Resources

Bibliography

Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability,1st edition,
www.ProbabilityBookstore.com, 2007.
Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory,2nd edition, World Scientific, 2006.
Geoffrey R. Grimmett, David R. Stirzaker, Probability and Random Processes,3rd edition, Oxford University
Press, 2001.
Richard Durrett, Probability: Theory and Examples,4th edition, Cambridge University Press, 2010.

Ressources en bibliothèque

- A Second Course in Probability / Ross
- Probability: Theory and Examples / Durrett
- Probability and Random Processes / Grimmett
- A First Look at Rigorous Probability Theory / Rosenthal

Notes/Handbook

available on the course website

Websites

- http://ipgold.epfl.ch/~leveque/Advanced_Prob/

Prerequisite for

Advanced classes requiring a good knowledge of probability

CS-437	**Algebraic coding theory**

Shokrollahi Mohammad Amin

Cursus	Sem.	Type
Information security minor	E	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

This course is a brief introduction to algebraic coding theory and is intended to provide the students with a basic understanding of methods for the construction, evaluation and application of algebraic error-correcting codes.

Content

- Linear Algebraic codes
- Evaluation codes
- Reed-Solomon codes and their (list)-decoding algorithms
- Efficient decoding: the displacement method
- Codes from algebraic geometry

Keywords

- Algebraic codes
- Reed-Solomon codes
- BCH codes
- Algebraic geometric codes
- List-decoding algorithms

Learning Prerequisites

Required courses

Obligatory courses of the first year in IN and SC

Recommended courses

Algebra for Communications

Important concepts to start the course

General mathematical maturity
Finite fields

Learning Outcomes

By the end of the course, the student must be able to:

- Design algebraic error-correcting codes with simple parameters
- Know encoding/decoding algorithms for these codes
- Have a basic understanding of evaluation codes and their applications
- Have a basic understanding of list-decoding algorithms

Teaching methods

Ex cathedra lectures

Expected student activities

Attendence at lectures and exercises

Assessment methods

Written exam
Only the grade on the final exam counts

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

Resources

Bibliography

- The Theory of Error-Correcting Codes by F.J. MacWilliams and N.J.A. Sloane
- Introduction to Coding Theory by J.H. van Lint

MATH-409   Algebraic curves and cryptography

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
Mathématiques pour l'enseignement	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Remarque

pas donné en 2017-18 - Cours donnés en alternance tous les deux ans

Summary

The goal of this course is to introduce basic notions from public-key cryptography based on algebraic curves over finite fields. We will introduce basic cryptographic schemes as well as discuss in-depth the discrete logarithm problem for elliptic and Jacobians of higher genus curves.

Content

Topics may include, but are not limited to:

- Introduction to algebraic curves
- Elliptic and hyperelliptic curves
- Jacobians of algebraic curves
- Cantor arithmetic
- Elliptic curve discrete logarithm problem
- Index calculus methods for Jacobians
- Pairing-based cryptography

Keywords

algebraic curves over finite fields, public key cryptography, discrete logarithms, pairing-based cryptography

Learning Prerequisites

Required courses

Abstract Algebra required (groups theory, rings, fields, field extensions, finite fields)

Recommended courses

- Math 317 (Galois theory)
- Math 489 (Number Theory in Cryptography)
- COM-401 (Security and Cryptography)

Teaching methods

Weekly lectures, problem sets and programming assignments.

Assessment methods

Weekly problem sets - 50% of the final grade
Programming assignements - 20% of the final grade
Final exam - 30% of the final grade

Supervision

Assistants   Yes

Resources

Bibliography

- P. Griffiths, *Introduction to Algebraic Curves*
- I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*
- I. Blake, G. Seroussi, N. Smart, *Advances in Elliptic Curve Cryptography*

CS-435   **Analytic algorithms**

Vishnoi Nisheeth

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

In the last decade, many fundamental algorithmic problems have benefited from viewing the underlying discrete problems through the lens of continuous/analytic methods. In this course we will introduce a selection of such techniques and explore their applications.

Content

Convexity and gradient descent
Multiplicative weight update (MWU) method and online convex optimization
Gradient descent based methods for solving linear equations
Optimization problems involving polynomials
Graphs and their eigenvectors and eigenvalues
Graphs as electrical networks
Graphs Laplacians and solving Laplacian equations
Application: Fast algorithms to compute network flows (using MWU, electrical flows and Laplacian solvers)
Application: Fast algorithms for graph cuts (using eigenvectors and Laplacian solvers)
Application: Algorithms for counting perfect matchings in graphs (using convex programs involving polynomials)

Keywords

Convex optimization, Spectral methods, Polynomials, Discrete Optimization, Continuous Optimization

Learning Prerequisites

Required courses

Calculus (MATH105), Linear Algebra (MATH110), Algorithms (CS250), Theory of Computation (CS251) or equivalents, Advanced Algorithms (CS-450) (or equivalent).

Recommended courses

Important concepts to start the course

This is an advanced course and requires mathematical maturity including linear algebra, multi-variate calculus, analysis, probability and algorithms.

Learning Outcomes

By the end of the course, the student must be able to:

- Learn fundamental techniques which apply continuous methods to discrete problems
- Apply analytic techniques to a variety of related problems
- Read, understand, and explain state of the art papers in this area

Assessment methods

Homeworks, Scribe Notes, Exam and Project/Presentation*.
*Tentative

Resources

Bibliography

Books relevant to the course:
Vishnoi - Lx=b
Nesterov - Introductory lectures on convex optimization
Shalev-Schwartz - Online learning and online convex optimization
References for Basics:
Apostol - Calculus I and II
Strang - Linear algebra and its applications
Boyd and Vanderberghe - Convex optimization
Strogatz - Nonlinear dynamics and Chaos

Ressources en bibliothèque

- Convex optimization / Boyd
- Linear algebra and its applications / Strang
- Nonlinear dynamics and Chaos / Strogatz
- Gaussian Hilbert Spaces / Janson
- Introductory lectures on convex optimization / Nesterov
- Mathematical view of interior point methods in convex optimization / Renegar
- Calculus I / Apostol
- Lx=b / Vishnoi
- Calculus II / Apostol

Notes/Handbook

Vishnoi - Zeros of Polynomials and their Applications to theory. Available from
http://theory.epfl.ch/vishnoi/Publications_files/ZerosIntro.pdf

Vishnoi -- A mini-course on convex optimization. Available from
http://theory.epfl.ch/vishnoi/Nisheeth-VishnoiFall2014-ConvexOptimization.pdf

MATH-474

Applied biostatistics

Goldstein Darlene

Cursus	Sem.	Type
Bioingénierie	MA2, MA4	Opt.
Data Science	MA2	Opt.
Informatique	MA2	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Mathématiques pour l'enseignement	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Sciences du vivant	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

This course covers topics in applied biostatistics, with an emphasis on practical aspects of data analysis using R statistical software. Topics include types of studies and their design and analysis, high dimensional data analysis (genetic/genomic) and other topics as time and interest permit.

Content

- Types of studies
- Design and analysis of studies
- R statistical software
- Reproducible research techniques and tools
- Report writing
- Exploratory data analysis
- High dimensional data analysis (genetics/genomics applications)
- Additional topics as time and interest permit

Evaluation is based on written reports of biostatistical data analyses.

Keywords

Data analysis, reproducible research, statistical methods, R, biostatistical data analysis, statistical data analysis

Learning Prerequisites

Required courses

This course will be very difficult for students with no previous course or experience with statistics. **Previous experience with R is neither assumed nor required.**

Recommended courses

Undergraduate statistics course

Learning Outcomes

By the end of the course, the student must be able to:

- Interpret analysis results
- Justify analysis plan
- Plan analysis for a given dataset

- Analyze various types of biostatistical data
- Synthesize analysis into a written report
- Report plan of analysis and results obtained

Transversal skills

- Write a scientific or technical report.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Take feedback (critique) and respond in an appropriate manner.
- Use a work methodology appropriate to the task.

Teaching methods

Lectures and practical exercises using R.

Expected student activities

Students are expected to participate in their learning by attending lectures and practical exercise sessions, posing questions, proposing topics of interest, peer reviewing of preliminary reports, and interacting with teaching staff regarding their understanding of course material.

Assessment methods

Evaluation is based on written reports of biostatistical data analyses.
Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

CS-401   **Applied data analysis**

West Robert

Cursus	Sem.	Type
Bioingénierie	MA1, MA3	Obl.
Data Science	MA1	Obl.
Génie électrique et électronique	MA1, MA3	Opt.
Humanités digitales	MA1	Obl.
Informatique	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Mineur en Neurosciences computationnelles	H	Obl.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

This course teaches the basic techniques and practical skills required to make sense out of a variety of data, with the help of the most acclaimed software tools in the data science world: pandas, scikit-learn, Spark, etc.

Content

Thanks to a new breed of software tools that allows to easily process and analyze data at scale, we are now able to extract invaluable insights from the vast amount of data generated daily. As a result, both the business and scientific world are undergoing a revolution which is fueled by one of the most sought after job profiles: the data scientist.
This course covers the fundamental steps of the data science pipeline:
Data Acquisition

- Variety as one of the main challenges in big data: structured, semi-structured, unstructured
- Data sources: open, public (scraping, parsing, and down-sampling)
- Dataset fusion, filtering, slicing & dicing
- Data granularities and aggregations

Data Wrangling

- Data manipulation, array programming, dataframes
- The many sources of data problems (and how to fix them): missing data, incorrect data, inconsistent representations
- Schema alignment, data reconciliation
- Data quality testing with crowdsourcing

Data Interpretation

- Stats in practice (distribution fitting, statistical significance, etc.)
- Co-occurrence grouping (market-basket analysis)
- Machine learning in practice (supervised and unsupervised, feature engineering, more data vs. advanced algorithms, curse of dimensionality, etc.)
- Text mining: vector space model, topic models, word embedding
- Social network analysis (influencers, community detection, etc.)

Data Visualization

- Introduction to different plot types (1, 2, and 3 variables), layout best practices, network and geographical data
- Visualization to diagnose data problems, scaling visualization to large datasets, visualizing uncertain data

Reporting

- Results reporting, infographics
- How to publish reproducible results
- Anonymiziation, ethical concerns

The students will learn the techniques during the ex-cathedra lectures, and will then get familiar with the software tools to complete the homework assignments (which will be in part executed under the supervision of the teacher and the assistants, during the lab hours).
In parallel, the students will embark in a semester-long project, split in agile teams of 3. The outcome of such team efforts will be unified towards the end of the course, to build a project portfolio that will be made public (and available as open-source).
At the end of the semester, students will also take a 3-hour final exam in a classroom with computers, where they will be asked to complete a data analysis pipeline (both with code and extensive comments) on a dataset they have never worked with before.

Keywords

data science, data analysis, data mining, machine learning

Learning Prerequisites

Required courses

The student MUST have passed an introduction to databases course, OR a course in probability & statistics, OR two separate courses that include programming projects.

Recommended courses

- CS-423 Distributed Information Systems
- CS-433 Pattern Classification and Machine Learning

Important concepts to start the course

Algorithms, object oriented programming, basic probability and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the Data Science pipeline
- Perform data acquisition (data formats, dataset fusion, Web scrapers, REST APIs, open data, big data platforms, etc.)
- Perform data wrangling (fixing missing and incorrect data, data reconciliation, data quality assessments, etc.)
- Perform data interpretation (statistics, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)

Transversal skills

- Give feedback (critique) in an appropriate fashion.

- Demonstrate the capacity for critical thinking
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

- Physical in-class recitations and lab sessions
- Homework assignments
- Course project

Expected student activities

Students are expected to:
- Attend the lectures and lab sessions
- Complete a weekly homework assignment
- Read/watch the pertinent material before a lecture
- Engage during the class, and present their results in front of the other colleagues

Assessment methods

- 30% continuous assessment during the semester (homework)
- 30% final exam, data analysis task on a computer (3 hours)
- 40% final project, done in groups of 3

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	http://ada.epfl.ch

Resources

Virtual desktop infrastructure (VDI)
No

Websites
- http://ada.epfl.ch

CS-456	**Artificial neural networks**		

Gerstner Wulfram

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique	MA2	Opt.
Mineur en Biocomputing	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Since 2010 approaches in deep learning have revolutionized fields as diverse as computer vision, machine learning, or artificial intelligence. This course gives a systematic introduction into the main models of deep artificial neural networks: Supervised Learning and Reinforcement Learning.

Content

- Simple perceptrons for classification
- BackProp and Multilayer Perceptron
- Deep Learning 1: Introduction
- Deep Learning 2: regularization and Tricks of the Trade
- Deep Learning 3: Theory
- Autoencoders and unsupervised learning
- Reinforcement Learning 1: TD Learning
- Reinforcement Learning 2: Q learning, SARSA
- Reinforcement Learning 3: Policy gradient
- Deep reinforcement learning
- Applications
- Outlook: Can the Brain implement Deep Learning?

Keywords

Deep learning, artificial neural networks, reinforcement learning, TD learning, SARSA,

Learning Prerequisites

Required courses

CS 433 Pattern Classification and Machine Learning (or equivalent)
Calculus, Linear Algebra (at the level equivalent to first 2 years of EPFL in STI or IC, such as Computer Science, Physics or Electrical Engineering)

Recommended courses

stochastic processes
optimization

Important concepts to start the course

- *Regularization in machine learning,*
- *Training base versus Test base, cross validation.*
- *Gradient descent. Stochastic gradient descent.*
- *Expectation, Poisson Process, Bernoulli Process.*

Learning Outcomes

By the end of the course, the student must be able to:

- Apply learning in deep networks to real data
- Assess / Evaluate performance of learning algorithms
- Elaborate relations between different mathematical concepts of learning
- Judge limitations of algorithms
- Propose algorithms and models for learning in deep networks

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Manage priorities.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.

Teaching methods

ex cathedra lectures and miniproject

Expected student activities

work on miniproject
attend all lectures
read book chapters and relevant tutorials
solve all exercises

Assessment methods

written exam (70 percent) and miniproject (3 percent)

Resources

Bibliography

- Textbook: Deep Learning by Goodfellow, Bengio, Courville
- Landmark papers

Links to videos of presentations given by people in deep learning

COM-415   **Audio signal processing and virtual acoustics**

Faller Christof, Kolundzija Mihailo

Cursus	Sem.	Type
Data Science	MA1	Opt.
Humanités digitales	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

The objective of the course is to introduce theory, methods, and basic psychoacoustics that is needed to understand state-of-the-art techniques used in pro audio and consumer audio, including microphones, surround sound, auralization, virtual acoustics, mixing, and audio coding.

Content

Acoustics and audio is covered and the manipulation and processing of audio signals. It is shown how Fourier analysis of the soundfield yields to the representation of a soundfield with plane waves. These and other acoustic insights are used to explain microphone techniques and reproduction of the soundfield.
Spatial hearing is covered in detail and used to motivate stereo and surround mixing and audio playback. In addition, insights on the principles of auralization and virtual acoustics are given, and the simulation of sound propagation in rooms will be further discussed.
The short-time Fourier transform is introduced as a tool for flexible manipulation of audio signals, suchs as filtering, delaying and other spectral modification. Matrix surround, audio coding, and beamforming are also treated.

Keywords

acoustics, virtual acoustics, microphones, surround sound, matrix surround, audio coding, audio processing, 3d sound reproduction, spatialization, psychoacoustics, human hearing, binaural hearing, dummy head recordings, wave propagation, simulation techniques, geometrical acoustics, auralization, sonification, audio, signal processing

Learning Prerequisites

Recommended courses

Fourier transform, signal processing basics (sampling, filtering, discrete Fourier transform).

Learning Outcomes

By the end of the course, the student must be able to:

• Apply Basics of Acoustics, Signal Processing, Reproduction, Simulation Techniques

• Implement Basics of Audio Signal Processing, Filtering, Multi-Channel Loudspeaker Setups

• Operate Room acoustics simulation programs

Teaching methods

Class + exercise sessions

Assessment methods

Midterm exam + Final exam

Resources

Bibliography

* C. Faller, "Signal Processing for Audio and Acoustics" complete lecture notes in book form.
* J. Blauert, "Spatial Hearing : The Psychophysics of Human Sound Localization", MIT Press, 2001.
* F. Rumsey, "Spatial Audio", Focal Press, 2001.
* M. Vorländer, "Auralization - Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality", 2010

Ressources en bibliothèque

• Spatial Hearing / Blauert
• Signal Processing for Audio and Acoustics / Faller
• Spatial Audio / Rumsey
• Auralization / Vorländer

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

EE-554   **Automatic speech processing**

Bourlard Hervé

Cursus	Sem.	Type
Data Science	MA1	Opt.
Génie électrique et électronique	MA1, MA3	Opt.
Humanités digitales	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

The goal of this course is to provide the students with the main formalisms, models and algorithms required for the implementation of advanced speech processing applications (involving, among others, speech coding, speech analysis/synthesis, and speech recognition).

Content

1. Introduction: Speech processing tasks, language engineering applications.

2. Basic Tools: Analysis and spectral properties of the speech signal, linear prediction algorithms, statistical pattern recognition, dynamic programming.

3. Speech Coding: Human hearing properties, quantization theory, speech coding in the temporal and frequency domains.

4. Speech Synthesis: Morpho-syntactic analysis, phonetic transcription, prosody, speech synthesis models.

5. Automatic Speech Recognition: Temporal pattern matching and Dynamic Time Warping (DTW) algorithms, speech recognition systems based on Hidden Markov Models (HMMs).

6. Speaker recognition and speaker verification: Formalism, hypothesis testing, HMM based speaker verification.

7. Linguistic Engineering: state-of-the-art and typical applications

Keywords

speech processing, speech coding, speech analysis/synthesis, automatic speech recognition, speaker identification, text-to-speech

Learning Prerequisites

Required courses

Basis in linear algebra, signal processing (FFT), and statistics

Important concepts to start the course

Basic knowledge in signal processing, linear algebra, statistics and stochastic processes.

Learning Outcomes

By the end of the course, the student must be able to:

- speech signal properties
- Exploit those properties to speech codign, speech synthesis, and speech recognition

Transversal skills

- Use a work methodology appropriate to the task.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

Teaching methods

Lecture + lab exercises

Expected student activities

Attending courses and lab exercises. Read additional papers and continue lab exercises at home if necessary. Regulary answer list of questions for feedback.

Assessment methods

Written exam without notes

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Ressources en bibliothèque

- Traitement de la parole / Boite

Websites

- http://lectures.idiap.ch/

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

BIO-465 **Biological modeling of neural networks**

Gerstner Wulfram

Cursus	Sem.	Type
Data Science	MA2	Opt.
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	MA2	Opt.
Mineur en Biocomputing	E	Opt.
Mineur en Neuroprosthétiques	E	Opt.
Mineur en Neurosciences computationnelles	E	Opt.
Mineur en Technologies biomédicales	E	Opt.
Neurosciences		Opt.
SC master EPFL	MA2, MA4	Opt.
Science et ing. computationelles	MA2	Opt.
Sciences du vivant	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

In this course we study mathematical models of neurons and neuronal networks in the context of biology and establish links to models of cognition.

Content

I. Models of single neurons 1. Introduction: brain vs computer and a first simple neuron model 2. Models on the level of ion current (Hodgkin-Huxley model) 3./4. Two-dimensional models and phase space analysis *II. Neuronal Dynamics of Cognition* 5./6. Associative Memory and Attractor Dynamics (Hopfield Model) 7. Neuronal Populations and networks 8. Continuum models and perception 9. Competition and models of Decision making *III. Noise and the neural code* 10. Noise and variability of spike trains (point processes, renewal process, interval distribution) 11: Variance of membrane potentials and Spike Response Models **IV. Plasticity and Learning** 12. Synaptic Plasticity and Long-term potentiation and Learning (Hebb rule, mathematical formulation) 13. Summary: Fitting Neural Models to Data

Keywords

neural networks, neuronal dynamics, computational neuroscience, mathematical modeling in biology, applied mathematics, brain, cognition, neurons, memory, learning, plasticity

Learning Prerequisites

Required courses

undergraduate math at the level of electrical engineering or physics majors
undergraduate physics.

Recommended courses

Analysis I-III, linear algebra, probability and statistics
For SSV students: Dynamical Systems Theory for Engineers or "Mathematical and Computational Models in Biology" course, Felix Naef

Important concepts to start the course

Differential equations, stochastic processes,

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze two-dimensional models in the phase plane
- Solve linear one-dimensional differential equations
- Develop a simplified model by separation of time scales
- Analyze connected networks in the mean-field limit
- Formulate stochastic models of biological phenomena
- Formalize biological facts into mathematical models
- Prove stability and convergence
- Apply model concepts in simulations
- Predict outcome of dynamics
- Describe neuronal phenomena

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Collect data.
- Write a scientific or technical report.

Teaching methods

Classroom teaching, exercises and miniproject

Expected student activities

miniprojects

Assessment methods

Written exam (67%) & miniproject (33%)

Resources

Bibliography

Gerstner, Kistler, Naud, Pansinski : Neuronal Dynamics, Cambridge Univ. Press 2014

Ressources en bibliothèque

- Neuronal Dynamics / Gerstner

Videos

- http://lcn.epfl.ch/~gerstner/VideoLecturesGerstner.html
- http://lcn.epfl.ch/~gerstner/NeuronalDynamics-MOOC1.html

EE-512   **Biomedical signal processing**

Vesin Jean-Marc

Cursus	Sem.	Type
Bioingénierie	MA1, MA3	Obl.
Génie électrique et électronique	MA1, MA3	Opt.
Informatique	MA1, MA3	Opt.
Mineur en Technologies biomédicales	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Project	2 weekly

Summary

The goal of this course is to introduce the techniques most commonly used for the analysis of biomedical signals, and to present concrete examples of their application for diagnosis purposes.

Content

1. Generalities on biomedical signal processing
2. Digital signal processing - basics

- sampling
- Fourier transform
- filtering
- stochastic signals correlation, and pwoer spectral density

3. Time-frequency analysis

- short-term Fourier transform
- time-frequency distributions, Cohen's class
- wavelet transform

4. Linear modeling

- autoregressive models
- linear prediction
- parametric spectral estimation
- criteria for model selection

5. Adaptive filtering

- adaptive predictione
- adaptive estimation of transfert functions
- adaptive interference cancellation

6. Miscellaneous

- polynomial models
- singular value decomposition
- principal component analysis

Keywords

signal processing, biomedical engineering, signal modeling, spectral analysis, adaptive filtering

Learning Prerequisites

Recommended courses

Signal processing for telecommunications COM-303
Signal processing EE-350

Teaching methods

lectures, lab sessions using Matlab

Assessment methods

1 point for lab/exercise sessions reports
2 exams: end of November 2points - final exam 3 points

Supervision

Office hours	Yes
Assistants	Yes

CS-490   **Business design for IT services**

Wegmann Alain

Cursus	Sem.	Type
Informatique	MA2	Opt.
Mineur STAS Chine	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly

Summary

We teach how to "design" an IT supported business initiative. We use insights from philosophy and psychology to concretely understand business models and analysis tools. Students work in groups on a project of their choice. Concrete fieldwork outside class and substantial readings are required.

Content

Individually, the students have to read the documents listed below. They make a synthesis of their contents. They need to apply the concepts presented in these documents on case studies and on their own project.

The students work, in groups, on a project. They:
(1) imagine a new (IT) service to develop,
(2) identify and analyze the relevant segments,
(3) validate their model with real customers and potential partners,
(4) define the qualitative and quantitative goals for the new (IT) service.

To represent their business idea, the students use Trade Your Mind - a web-based business modelling service,

Keywords

Business services, IT services, business design, innovation in startups, revolutionary ventures and corporate initiatives; entrepreneur profiles.

Business design, service design, house of quality, SEAM modeling (eco-system, supplier-adopter relationship, motivation models)

Segmentation, value networks, PESTLE analysis, 5 forces analysis, core competency, coopetition, blue ocean, resource based modeling, transaction cost.
Integrated marketing concept, SWOT analysis, strategy canvas.
New technology adoption, crossing-the chasm, decision making units.
Pricing strategy, cashflow management, break-event time
Psychological types, epistemology, ontology, axiology (ethics and aesthetics).

Learning Outcomes

By the end of the course, the student must be able to:
- Create a precise and detailed description for a new business design
- Analyze environmental as well as organizational factors in a business design
- Design a business model in details (ecosystem, value, finance)
- Assess / Evaluate alternative business and technical strategies
- Synthesize multiple marketing theories (from seminal publications)
- Represent the key concepts of a business design (ecosystem, value, finance)

- Interpret evidences
- Investigate innovative views of a business design

Transversal skills

- Collect data.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Make an oral presentation.
- Summarize an article or a technical report.

Teaching methods

Problem-based teaching + group work

Resources

Bibliography

Bhide, A. (2000). The Origin and Evolution of New businesses: Oxford University Press.

Hauser, J. R., & Clausing, D. (1988). The house of Quality. Harvard Business Review.
https://hbr.org/1988/05/the-house-of-quality

Golnam, A., Regev, G., Ramboz, J., Laprade, P., & Wegmann, A. (2011). Aligning Value and Implementation in Service Design - A Systemic Approach. International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 3(1), 19-36.

Porter, M. E. (2008). The Five Competitive Forces That Shape Strategy. Harvard Business Review.
https://hbr.org/2008/01/the-five-competitive-forces-that-shape-strategy

Levitt, T. (1960). Marketing Myopia. Harvard Business Review.
https://hbr.org/2004/07/marketing-myopia

Prahalad, C., & Hamel, G. (1990). The Core Competence of the Corporation. Harvard Business Review.
https://hbr.org/1990/05/the-core-competence-of-the-corporation

Brandenburger, A. M., & Nalebuff, B. J. (1995). The Right Game: Use Game Theory to Shape Strategy. Harvard Business Review.
https://hbr.org/1995/07/the-right-game-use-game-theory-to-shape-strategy

Hagel, J., & Singer, M. (1999). Unbundling the Corporation. Harvard Business Review.
https://hbr.org/1999/03/unbundling-the-corporation

Kim, W. C., & Mauborgne, R. (2004). Blue Ocean Strategy, Havard Business Review.
https://hbr.org/2004/10/blue-ocean-strategy

Tools: Trade Your Mind – Business modeling tool on the web
www.tradeyourmind.com
Note: the list is non-exhaustive.

Ressources en bibliothèque

- A Framework for Modeling Value in Service-Oriented Business Models / Golnam
- The origin and evolution of new businesses / Bhide
- The structure of "unstructured" decision processes / Mintzberg

- Value Map / Golnam
- A Modeling Framework for Analyzing the Viability of Service Systems / Golnam
- Unbundling the corporation / Hagel
- Blue Ocean Strategy / Kim
- Competitive advantage / Porter
- Marketing myopia / Levitt
- The core competence of the corporation / Prahalad
- Aligning Value and Implementation in Service Design / Golnam
- The house of quality / Hauser
- Coopetition within andbetween value networks / Golnam

BIO-105   **Cellular biology and biochemistry for engineers**

Hirling Harald

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
Mineur en Neuroprosthétiques	H	Opt.
Mineur en Technologies biomédicales	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

Basic course in biochemistry as well as cellular and molecular biology for non-life science students enrolling at the Master or PhD thesis level from various engineering disciplines. It reviews essential notions necessary for a training in biology-related engineering fields.

Content

The course gives basic knowledge on various phenoma taking place within a cell, and among cells within tissues and organs. The course gives an integrated view of various molecular mechanisms (rather in the second half of the class). It should therefore allow engineering students involved in future projects touching on biomedical problems to better integrate the constraints of a biological system and to enable them to communicate with specialists in both fields.
This course is not available to students who had already taken basic cell biology or biochemistry classes during their Bachelor studies at EPFL or elsewhere. This applies for example to the course BIO-109 "Introduction to Life Sciences for Information Sciences".

Keywords

The course contains chapters on the following subjects:
1.Cells and Organs
2.Chemical components of cells
3.Proteins, Enzymes
4.Energy, Metabolism
5.DNA, Chromosomes, Replication
6.Gene expression
7.Recombinant techniques
8.Membrane and Transport
9.Intracellular trafficking
10.Cytoskeleton
11.Cell division, Mitosis
12.Genetics, Meiosis
13.Cell communication, Signaling
14.Tissue, Tissue regeneration

Learning Prerequisites

Required courses

Bachelor degree in engineering or other non-life science discipline

Recommended courses

Some basic knowledge in chemistry can help, but not required

Important concepts to start the course

Curiosity about how biological systems work, willingness to acquire a certain amount of facts and details necessary to understand and discuss the various molecular mechanisms present in cells or related to modern biology

Learning Outcomes

By the end of the course, the student must be able to:
- Describe the basic components and functions found in cells
- Draw schemes explaining essential cellular phenomena
- Explain which are the important metabolic pathways
- Translate information from genetic code
- Verify statements about specific cellular mechanisms
- Integrate knowledge from different cellular mechanisms

Transversal skills

- Access and evaluate appropriate sources of information.

Teaching methods

2 hours of ex cathedra-type of lecture
2 hours of exercises: the instructor gives out appr. 10 questions out (through Moodle and in the beginning of the session). The questions have different formats, and can in some cases just retrieve the acquired facts, in others have a more integrative problem-based learning approach.

Expected student activities

- review regularly the presented lectures.
- participate actively in the exercise sessions when the questions and problems are discussed altogether

Assessment methods

- a blank exam is performed around early December (does not give credits or bonus)
- a written exam at the winter exam session

Supervision

Office hours	Yes
Assistants	No
Forum	No
Others	- the teacher can always be reached through Email or phone to fix a one-to-one discussion about specific subjects
	- whether assistants will be involved depends on the number of students registered

Resources

Bibliography

The lecture is aligned to selected chapters in the following book (recommended although not required): "Essential Cell Bioogy" by B Alberts et al. , 3rd edition, Garland Science Taylor & Francis Group

Ressources en bibliothèque

- Essential Cell Biology / Alberts

CS-413   **Computational photography**
                    Süsstrunk Sabine

Cursus	Sem.	Type
Data Science	MA2	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

The students will gain the theoretical knowledge in computational photography, which allows recording and processing a richer visual experience than traditional digital imaging. They will also execute practical group projects to develop their own computational photography application.

Content

Computational photography is the art, science, and engineering of creating a great (still or moving) image. Information is recorded in space, time, across visible and invisible radiation and from other sources, and then post-processed to produce the final - visually pleasing - result.
Basics: Human vision system, Light and illumination, Geometric optics, Color science, Sensors, Digital camera systems.
Generalized illumination: Structured light, High dynamic range (HDR) imaging, Time-of-flight.
Generalized optics: Coded Image Sensing, Coded aperture, Focal stacks.
Generalized sensing: Low light imaging, Depth imaging, Plenoptic imaging, Light field cameras.
Generalized processing: Super-resolution, In-painting, Compositing, Photomontages, Panoramas, HDR imaging, Multi-wavelength imaging, Dynamic imaging.
Generalized display: Stereoscopic displays, HDR displays, 3D displays, Mobile displays.

Keywords

Computational Photography, Coded Image Sensing, Non-classical image capture, Multi-Image & Sensor Fusion, Mobile Imaging.

Learning Prerequisites

  Required courses

 • A basic Signal Processing, Image Processing, and/or Computer Vision course.
 • Linear Algebra.

  Recommended courses

 • Introduction to Computer Vision.
 • Signal Processing for Communications.

  Important concepts to start the course

 • Basic signal processing.

- Basic computer vision.
- Basic programming (iOS, Android, Matlab).

Learning Outcomes

- Identify the main components of a computational photography system.
- Contextualise the main trends in computational optics, sensing, processing, and displays.
- Create a computational photography application on a mobile platform.
- Design a computational photography solution to solve a particular imaging task.
- Assess / Evaluate hardware and software combinations for their imaging performance.
- Formulate computational photography challenges that still need to be resolved.

Transversal skills

- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

The course consists of 2 hours of lectures per week that will cover the theoretical basics. An additional 2 hours per week are dedicated to a group project designing, developing, and programming a compuational photography application on a mobile plateform (iOS, Android).

Expected student activities

The studens is expected to attend the class and actively participate in the practical group project, which requires coding on either Android or iOS plateform. The student is also required to read the assigned reading material (book chapters, scientific articles).

Assessment methods

The theoretical part will be evaluated with an oral exam at the end of the semester, and the practical part based on the students' group projects.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Selected book chapters
- Course notes (on moodle)
- Links to relevant scientific articles and on-line resources will be given on moodle.

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CS-442 · **Computer vision**
Fua Pascal

Cursus	Sem.	Type
Data Science	MA2	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
Mineur en Informatique	E	Opt.
Mineur en Systèmes de communication	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Computer Vision aims at modeling the world from digital images acquired using video or infrared cameras, and other imaging sensors. We will focus on images acquired using digital cameras. We will introduce basic processing techniques and discuss their field of applicability.

Content

Introduction
- History of Computer Vision
- Human vs Machine Vision
- Image formation

Extracting 2D Features
- Contours
- Texture
- Regions

3D Shape Recovery
- From one single image
- From multiple images

Learning Prerequisites

Recommended courses

Foundations of Image Science

Learning Outcomes

By the end of the course, the student must be able to:
- Choose relevant algorithms in specific situations
- Perform simple image-understanding tasks

Teaching methods

Ex cathedra lectures and programming exercises using matlab.

Assessment methods

With continuous control

Resources

Bibliography

• R. Szeliski, Computer Vision: Computer Vision: Algorithms and Applications, 2010.

• A. Zisserman and R. Hartley, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

Ressources en bibliothèque

• Computer Vision: Algorithms and Applications / Szeliski

• Multiple View Geometry in Computer Vision / Zisserman

Websites

• http://cvlab.epfl.ch/

Moodle Link

• http://moodle.epfl.ch/course/view.php?id=472

CS-453 | **Concurrent algorithms**

Guerraoui Rachid

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

With the advent of multiprocessors, it becomes crucial to master the underlying algorithmics of concurrency. The objective of this course is to study the foundations of concurrent algorithms and in particular the techniques that enable the construction of robust such algorithms.

Content

Model of a parallel system
A Multicore architect
Processes and objects
Safety and liveness

Parallel programming
Automatic parallelism
Mutual exclusion and locks
Non-blocking data structures

Register Implementations
Safe, regular and atomic registers
General and limited transactions
Atomic snapshots

Hierarchy of objects
The FLP impssibility
The consensus number
Universal constructions
Transactional memories
Transactional algorithms
Opacity and obstruction-freedom

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites

Required courses
ICC, operatings systems

Recommended courses
Algorithms, concurrency

Important concepts to start the course

Processes, threads, datas structures

Learning Outcomes

By the end of the course, the student must be able to:
- Reason in a precise manner about concurrency
- Design a concurrent algorithm

Teaching methods

Lectures and exercises

Expected student activities

Attendance at lectures completing exercise and sometimes doing a project

Assessment methods

With continuous control, mid-term final exams and sometimes project

Supervision

Office hours   Yes
Assistants   Yes
Forum   No

Resources

Websites

- http://lpd.epfl.ch/site/education

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CS-454   Convex optimization and applications

Lebret Hervé

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique	MA2	Opt.
Mineur en Systems Engineering	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Science et ing. computationelles	MA2	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	1 weekly
Exercises	2 weekly

Summary

Optimization is not only a major segment of applied mathematics, it is also a critical problem in many engineering and economic fields. In any situation where resources are limited, decision makers try to solve problems they face in the best possible manner. The course provides theory and practice.

Content

The class will cover topics such as:
Convex sets and functions
Recognizing convex optimization problems
Optimality Conditions and Duality
Linear Programming (geometry of linear programming, applications in network optimization, the simplex method)
Least squares and quadratic programs
Semidefinite programming
Interior point methods

Keywords

Convex Optimisation

Learning Prerequisites

Required courses

A good background in linear algebra. Mastering MATLAB is a plus!

Recommended courses

Basic Linear Algebra

Learning Outcomes

By the end of the course, the student must be able to:
• Solve Convex optimization problems

Teaching methods

Ex-cathedra lectures and exercise sessions(in English).

Assessment methods

Midterm (25%) and final exam (50%). Small personal project (25%). Exams are open-text and on paper (no use of computers)

Resources

Bibliography

Book : Convex Optimization by Stephen Boyd and Lieven Vandenberghe

Ressources en bibliothèque

• Convex Optimization / Boyd

COM-401   **Cryptography and security**
Vaudenay Serge

Cursus	Sem.	Type
Data Science	MA1	Opt.
Information security minor	H	Opt.
Informatique et communications		Obl.
Informatique	MA1, MA3	Obl.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

This course introduces the basics of cryptography. We review several types of cryptographic primitives, when it is safe to use them and how to select the appropriate security parameters. We detail how they work and sketch how they can be implemented.

Content

1. **Ancient cryptography**: Vigenère, Enigma, Vernam cipher, Shannon theory
2. **Diffie-Hellman cryptography**: algebra, Diffie-Hellman, ElGamal
3. **RSA cryptography**: number theory, RSA, factoring
4. **Elliptic curve cryptography**: elliptic curves over a finite field, ECDH, ECIES
5. **Symmetric encryption**: block ciphers, stream ciphers, exhaustive search
6. **Integrity and authentication**: hashing, MAC, birthday paradox
7. **Applications to symmetric cryptography**: mobile telephony, Bluetooth, WiFi
8. **Public-key cryptography**: cryptosystem, digital signature
9. **Trust establishment**: secure communication, trust setups
10. **Case studies**: Bluetooth, TLS, SSH, PGP, biometric passport

Keywords

cryptography, encryption, secure communication

Learning Prerequisites

Required courses

- Algebra (MATH-310)
- Probability and statistics (MATH-310)
- Algorithms (CS-250)

Recommended courses

- Network security (COM-301)

Important concepts to start the course

- Mathematical reasoning
- Probabilities
- Algebra, arithmetics

• Algorithmics

Learning Outcomes

By the end of the course, the student must be able to:

• Choose the appropriate cryptographic primitive in a security infrastructure
• Judge the strength of existing standards
• Assess / Evaluate the security based on key length
• Implement algorithms manipulating big numbers and use number theory
• Use algebra and probability theory to analyze cryptographic algorithms
• Identify the techniques to secure the communication and establish trust

Teaching methods

ex-cathedra

Expected student activities

• active participation during the course
• take notes during the course
• do the exercises during the exercise sessions
• complete the regular tests and homework
• read the material from the course
• self-train using the provided material
• do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

• homework (30%)
• regular graded tests (30%)
• midterm exam (40%)

Final exam averaged (same weight) with the contiuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

• Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
• A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press 2005.

Ressources en bibliothèque

- Communication security / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- http://lasec.epfl.ch/teaching.shtml

Prerequisite for

- Advanced cryptography (COM-401)
- Algorithms in public-key cryptography (COM-408)

COM-480   **Data visualization**

Benzi Kirell Maël

Cursus	Sem.	Type
Data Science	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

Understanding why and how to present complex data interactively in an effective manner has become a crucial skill for any data scientist. In this course, you will learn how to design, judge, build and present your own interactive data visualizations.

Content

1. Introduction
2. The Web : languages, tool, librairies
a. Basics (environnement, tools
b. HTML5, Javascript, DOM
c. D3.js
d. Basic charts
3. Visualization fundamentals
a. Human perception, user experience
b. Data types
c. Marks & Channels
d. Color theory
e. Methodology for designing a data-viz
4. Visualizing data, algorithms
a. Multivariate data
b. Maps
c. Trees
d. Networks
e. Volumes
5. Case studies

Keywords

Data viz, visualization, data science

Learning Prerequisites

Required courses

CS-305 Software engineering (BA)
CS-250 Algorithms (BA)
CS-401 Applied data analysis (MA)

Recommended courses
EE-558 A Network Tour of Data Science (MA)

CS-486 Human computer interaction (MA)
CS-210 Functional programming (BA)

Important concepts to start the course

Knowledge of one of the following progrmaming language such as C++, Python, Scala.
Familiarity with web-development (you already have a blog, host a webiste). Experience with THML5,
Javascript is a strong plus for the course.

Learning Outcomes

By the end of the course, the student must be able to:

• Judge visualization in a critical manner and suggest improvements.

• Design and implement visualizations from the idea to the final product according to human perception and cognition

• Know the common data-viz techniques for each data domain (multivariate data, networks, texts, cartography, etc) with their technical limitations

• Create interactive visualizations int he browser using HTM5 and Javascript

Teaching methods

Ex cathedra lectures, exercises, and group projects.

Expected student activities

• Follow lectures

• Read lectures notes, and textbooks

• Do an oral presentation of an original data-viz found on the web

• Create an advance data-viz in groups (group project)

• Write a series of blog post on the creation of the data-viz (group project)

Assessment methods

• Oral presentation of data-viz found on the web (10%)

• Group project data-viz (50%)

• Written report on the group project as a series of blog posts (40%)

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Bibliography

Visualization Analysis and Design by Tamara Munzner, CRC Press (2014). Fee online version at EPFL.
Interactive Data Visualization for the Web by Scott Murray O'Reilly (2013) - D3 - Free online version.

Notes/Handbook

Lecture notes

CS-422   **Database systems**

Ailamaki Anastasia

Cursus	Sem.	Type
Humanités digitales	MA2	Opt.
Informatique et communications		Opt.
Informatique	MA2	Obl.
Mineur en Informatique	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	During the semester
Workload	210h
Weeks	14
Hours	**7 weekly**
Lecture	3 weekly
Exercises	2 weekly
Project	2 weekly

Summary

This course is intended for students who want to understand modern large-scale data analysis systems and database systems. It covers a wide range of topics and technologies, and will prepare students to be able to build such systems as well as read and understand recent research publications.

Content

- Database systems
- Online analytics; data stream processing
- Column stores
- Decision support systems and data warehouses
- Large-scale data analytics infrastructure and systems
- Transaction processing. OLTP systems and concurrency control algorithms
- Distributed data management systems
- Query optimization; database tuning
- Logging and recovery
- Modern storage hierarchies

Learning Prerequisites

Required courses

- CS-322: Introduction to database systems
- CS-105: Introduction to object-oriented programming

Recommended courses

- CS-323: Operating systems
- CS-452: Foundations of software

Learning Outcomes

By the end of the course, the student must be able to:
- Design big data analytics systems using state-of-the-art infrastructures for horizontal scaling, e.g., Spark

• Implement algorithms and data structures for streaming data analytics

• Decide between different storage models based on the offered optimizations enabled by each model and on the expected query workload

• Compare concurrency control algorithms, and algorithms for distributed data management

• Identify performance culprits, e.g., due to concurrency control

Teaching methods

Lectures, project, homework, exercises

Expected student activities

During the semester, the students are expected to:

• attend the lectures in order to ask questions and interact with the professor,

• attend the exercise sessions to solve and discuss exercises about the recently taught material,

• work on projects, which cover the practical side of the taught material,

• take a midterm,

• take a final exam,

• read scientific papers related to the course material

Assessment methods

• 70% exams

• 30% project

Supervision

Office hours	Yes
Others	Office hours on request. Q&A sessions in lectures and exercises.

Resources

Bibliography

J. Hellerstein & M. Stonebraker, Readings in Database Systems, 4th Edition, 2005
R. Ramakrishnan & J. Gehrke: "Database Management Systems", McGraw-Hill, 3rd Edition, 2002.
A. Rajaraman & J. Ullman: "Mining of Massive Datasets", Cambridge Univ. Press, 2011.

CS-438   **Decentralized systems engineering**

Ford Bryan Alexander

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly

Summary

A decentralized system is one that works when no single party is in charge or fully trusted. This course teaches decentralized systems principles while guiding students through the development and testing of their own decentralized system incorporating messaging, encryption, and blockchain concepts.

Content

• Networking Foundations. Addressing, forwarding, routing. Client/server versus peer-to-peer communication. Firewalls, NATs, traversal.

• Gossip: a foundation for decentralized systems. UseNet: technical, security, and social lessons. Randomized rumor-mongering and anti-entropy algorithms.

• Communicating Securely: Basic Cryptographic Tools. Symmetric-key encryption. Hash functions, message authentication. Diffie-Hellman key exchange. Public-key encryption, digital signatures.

• Trust and Reputation. Authorities, trust networks. Sybil attacks and defenses.

• Naming and search. Request flooding. Hierarchical directories, landmark structures. Self-certifying identities. Distributed hash tables.

• Distributed consensus, distributed ledgers (blockchains), and cryptocurrencies.

• Following a Moving Target. Location services, reference points, forwarding. Composite identities.

• Anonymous Communication. Onion routing, mix networks. Dining cryptographers. Voting, verifiable shuffles, homomorphic encryption. Anonymous disruption.

• Fireproofing Alexandria: Decentralized Storage. Replication. Parity, erasure coding. Renewal. Digital preservation.

• Content Distribution. Opportunistic caching (FreeNet). Content integrity: hash trees, hash file systems. Convergent encryption. Swarming downloads: BitTorrent. Free-riding, incentives.

• Gaining perspective. Spam, malicious content. Review/moderation and reputation systems. Leveraging social networks (Peerspective). Balancing local and global viewpoints.

• Decentralized Collaboration. Network file systems, version management. Consistency.

• Pessimistic locking. Disconnected operation, eventual consistency, conflict resolution.

• Distributed Consensus. Paxos. Accountability (PeerReview). Byzantine fault tolerance.

• Mobile Code and Agents. Privacy: trusted computing, fully homomorphic encryption. Decentralized virtual organizations.

Keywords

distributed systems, decentralized systems, security, privacy, anonymity, cryptography, gossip, consensus, swarming, blockchain, cryptocurrency

Learning Prerequisites

Required courses

- COM-208: Computer networks

Recommended courses

- CS-206 Parallelism and concurrency
- COM-301: Network security
- CS-323: Introduction to operating systems

Important concepts to start the course

Students must already be highly competent at programming and debugging in a high-level systems programming language such as Java, C#, or Go. Programming exercises will be in Go, but students already well-versed and experienced in programming with comparable systems languages should be able to pick up Go during the course.

Students should have both solid foundational knowledge of how networks function, and some experience actually writing network programs, e.g., TCP/IP programming using the Sockets API.

Learning Outcomes

By the end of the course, the student must be able to:

- Design practical distributed and decentralized systems
- Implement systems via hands-on coding, debugging, and interoperability testing

Teaching methods

Lectures: The course's lectures will present and discuss challenges, known techniques, and open questions in decentralized system design and implementation. Lectures will often be driven by examination of real decentralized systems with various purposes in widespread use the past or present, such as UseNet, IRC, FreeNet, Tor, BitTorrent, and Bitcoin. Throughout the course we will explore fundamental security and usability challenges such as decentralized identification and authentication, denial-of-service and Sybil attacks, and maintenance of decentralized structures undergoing rapid changes (churn).

Labs: During the semester, students will develop a small but usable peer-to-peer communication application that reflects a few of the important design principles and techniques to be explored in the course, such as gossip, social trust networks, distributed hash tables, consensus algorithms, and cryptocurrencies. The labs will designed so that solutions can initially be tested individually on private, virtual networks running on one machine, then tested collectively by attempting to make different students' solutions interoperate on a real network.

Expected student activities

Students will be expected to attend lectures to understand the concepts needed for the course, but the main workload will be actual hands-on programming assignments, which the students will perform on their own during the first part of the course and optionally in small teams during the final project-oriented part of the course.

Assessment methods

- Lecture attendance: 10%

- Programming assignment grading (evaluating both function and student documentation): 50%

- Final project grading (accounting for both scope, appropriateness, and follow-through in implementation quality and documentation): 40%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

CS-472   Design technologies for integrated systems

De Micheli Giovanni

Cursus	Sem.	Type
Computer engineering minor	H	Opt.
Génie électrique et électronique	MA1, MA3	Opt.
Informatique et communications		Opt.
Informatique	MA1, MA3	Opt.
Mineur STAS Chine	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Project	2 weekly

Summary

Hardware compilation is the process of transforming specialized hardware description languages into circuit descriptions, which are iteratively refined, detailed and optimized. The course presents algorithms, tools and methods for hardware compilation and logic synthesis.

Content

The course will present the most outstanding features of hardware compilation, as well as the techniques for optimizing logic representations and networks. The course gives a novel, uptodate view of digital circuit design. Practical sessions will teach students the use of current design tools.Syllabus1) Modeling languages and specification formalisms;2) High-level synthesis and optimization methods (scheduling, binding, data-path and control synthesis);3) Representation and optimization of combinational logic functions (encoding problems, binary decision diagrams);4) Representation and optimization of multiple-level networks (algebraic and Boolean methods, "don't care" set computation, timing verification and optimization);5) Modeling and optimization of sequential functions and networks (retiming);6) Semicustom libraries and library binding.

Keywords

Hardware, VLSI, Synthesis, Optimization, Algorithms

Learning Prerequisites

Required courses

No specific course

Recommended courses

Knowledge of digital design, algorithm design and programming.

Important concepts to start the course

Knowledge of digital design, algorithm design and programming.

Learning Outcomes

By the end of the course, the student must be able to:
- Recognize important problems in digital design
- Examine and evaluate available design tools and methods
- Decide upon a design tool flow to perform a digital design

Transversal skills

• Plan and carry out activities in a way which makes optimal use of available time and other resources.

Assessment methods

Continuous control :
Homework : 30 %, Project 15 %, Midterm test : 20 %,
End term test : 35 %

Resources

Virtual desktop infrastructure (VDI)
No

Bibliography
G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw'Hill.

Ressources en bibliothèque
• Synthesis and Optimization of Digital Circuits / De Micheli

Notes/Handbook
Copies of the slides used for lectures will be given in class and posted.

Websites
• http://lsi-www.epfl.ch/dtis/

CS-446

Digital 3D Geometry Processing

Pauly Mark

Cursus	Sem.	Type
Humanités digitales	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly

Summary

Students study & apply core concepts and algorithms for digital geometry processing & 3D content creation. They create their own digital and physical geometry in a group project that follows the digital 3D content creation pipeline from data acquisition, geometry processing, to physical fabrication.

Content

The course will follow the digital 3D content creation pipeline. We will first discuss the fundamentals of geometry representations and cover continuous and discrete differential geometry concepts. Polygon mesh representations will be at the center of our investigations. We derive the core processing methods for triangle meshes, such as surface smoothing, parameterization, remeshing or deformation. Besides the mathematical concepts and algorithmic foundations, the course puts strong emphasis on implementation and features an extensive project. For the project, students will scan their own 3D models, edit and enhance them with geometry processing algorithms, and finally map their geometric models to digital fabrication processes (3D printing, laser cutting) to create physical realizations of their models.

Keywords

geometry, 3D modeling, polygon meshes, digital fabrication

Learning Prerequisites

Required courses

Linear Algebra, Calculus, Programming

Recommended courses

Introduction to Computer Graphics

Learning Outcomes

By the end of the course, the student must be able to:

- Explain and contrast fundamental geometry representations
- Explain and apply basic concepts from discrete differential geometry
- Analyze the 3D content creation pipeline and understand its limitations
- Implement and evaluate basic geometry processing algorithms, such as smoothing, remeshing, deformation, and constructive solid geometry
- Create digital 3D models from photographs and process the acquired raw geometry to build physical prototypes
- Coordinate a team during a software project

Teaching methods

Lectures, interactive demos, theory and programming exercises, programming project, project tutoring

Expected student activities

The student are expected to study the provided reading material and actively participate in class. They should prepare and resolve the exercises, prepare and carry out the programming project. Exercises in the first half of the course are done in groups of three students. For the second half of the course, the project is done in larger teams.

Assessment methods

Exercises, project, final examination

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

A list of books will be provided at the beginning of the class

Ressources en bibliothèque

• Polygon Mesh Processing / Botsch

Notes/Handbook

Slides and online resources will be provided in class

Websites

• http://lgg.epfl.ch/DGP

CS-411   **Digital education & learning analytics**

Dillenbourg Pierre, Jermann Patrick

Cursus	Sem.	Type
Data Science	MA1	Opt.
Humanités digitales	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Oral
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

This course addresses the relationship between specific technological features and the learners' cognitive processes. It also covers the methods and results of empirical studies on this topic: do student actually learn due to technologies?

Content

Learning theories and learning processes. Instructional design: methods, patterns and principles. Orchestration graphs. On-line education. Effectiveness of learning technologies. Methods for empirical research. Learning analytics. History of learning technologies.

Keywords

learning, pedagogy, teaching, online education, MOOCs

Learning Prerequisites

Recommended courses

One of these courses is recommended:
- Machine Learning (Jaggi / Urbanke)
- Applied Data Analysis (West)

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the learning processes triggered by a technology-based activity
- Explain how a technology feature influences learning processes
- Elaborate a study that measures the learning effects of a digital environment
- Select appropriately a learning technology given the target audience and the expected learning outcomes
- Apply machine learning methods to educational traces

Transversal skills

- Set objectives and design an action plan to reach those objectives.

Teaching methods

The course will combine participatory lectures with a project around learning analytics

Expected student activities

The project will include several milestones to be delivered along the semester.

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14248

CS-451	**Distributed algorithms**

Guerraoui Rachid

Cursus	Sem.	Type
Informatique	MA1, MA3	Obl.
Mineur en Informatique	H	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Computing is often distributed over several machines, in a local IP-like network, a cloud or in a P2P network. Failures are common and computations need to proceed despite partial failures of machines or communication links. The foundations of reliable distributed computing will be studied.

Content

Reliable broadcast
Causal Broadcast
Total Order Broadcast
Consensus
Non-Blocking Atomic Commit
Group Membership, View Synchrony
Terminating Reliable Broadcast
Shared Memory in Message Passing System
Byzantine Fault Tolerance
Self Stabilization
Population protocols   (models of mobile networks)

Keywords

Distributed algorithms, checkpointing, replication, consensus, atomic broadcast, ditributed transactions, atomic commitment, 2PC.

Learning Prerequisites

Required courses

Basics of Algorithms, networking and operating systems

Recommended courses

The lecture is orthogonal to the one on concurrent algorithms: they can be taken in parallel.

Learning Outcomes

By the end of the course, the student must be able to:
- Choose an appropriate abstraction to model a distributed computing problem
- Specify the abstraction
- Present an implementation of it
- Analyze its complexity

Teaching methods

Ex cathedera

Assessment methods

Mid-term and final exams.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Ressources en bibliothèque

- Introduction to reliable and secure distributed programming / Cachin

Notes/Handbook

Reliable and Secure Distributed Programming
Springer Verlag
C. Cachin, R. Guerraoui, L. Rodrigues

Websites

- http://lpdwww.epfl.ch/education

Videos

- http://wandida.com

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CS-423 **Distributed information systems**

Aberer Karl

Cursus	Sem.	Type
Data Science	MA2	Opt.
Energie et durabilité	MA2, MA4	Opt.
Génie électrique et électronique	MA2, MA4	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Obl.
Mineur en Biocomputing	E	Obl.
Mineur en Informatique	E	Opt.
Mineur en Systèmes de communication	E	Opt.
SC master EPFL	MA2, MA4	Obl.
Sciences et ingénierie de l'environnement	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

This course introduces in detail several key technologies underlying today's distributed information systems, including Web data management, information retrieval and data mining.

Content

Web Information Management: Semi-structured data - graph data model, web ontologies, schema integration

Information Search: Web search - vector space retrieval, inverted files, advanced retrieval models, word embeddings, web search

Big Data Analytics: Data mining - associations rules, clustering, classification, model selection; Crowd-sourcing; Recommender systems - collaborative filtering and content-based recommendation

Learning Prerequisites

Recommended courses

Introduction to Database Systems

Learning Outcomes

By the end of the course, the student must be able to:

• Characterize the main tasks performed by information systems, namely data, information and knowledge management

• Apply semi-structured data models, their representation through Web standards and algorithms for storing and processing semi-structured data

• Apply fundamental models and techniques of text retrieval and their use in Web search engines

• Apply main categories of data mining techniques, local rules, predictive and descriptive models, and master representative algorithms for each of the categories

• Apply collaborative information management models, like crowd-sourcing, recommender systems, social networks

Teaching methods

Ex cathedra + exercises

Assessment methods

25% Continuous evaluations with bonus system during the semester
75% Final written exam (180 min) during exam session

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Websites

- http://lsir.epfl.ch/teaching/current-courses/

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=4051

ENG-466   **Distributed intelligent systems**

Martinoli Alcherio

Cursus	Sem.	Type
Computer engineering minor	H	Opt.
Data Science	MA1	Opt.
Energie et durabilité	MA1, MA3	Opt.
Informatique	MA1, MA3	Opt.
Microtechnique	MA1, MA3	Opt.
Mineur en Biocomputing	H	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Sciences et ingénierie de l'environnement	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	3 weekly

Summary

The goal of this course is to provide methods and tools for modeling distributed intelligent systems as well as designing and optimizing coordination strategies. The course is a well-balanced mixture of theory and laboratory exercises using simulation and real hardware platforms.

Content

- Introduction to key concepts such as self-organization and software and hardware tools used in the course
- Examples of natural, artificial and hybrid distributed intelligent systems
- Modeling methods: sub-microscopic, microscopic, macroscopic, multi-level; spatial and non-spatial; mean field, approximated and exact approaches
- Machine-learning methods: single- and multi-agent techniques; expensive optimization problems and noise resistance
- Coordination strategies and distributed control: direct and indirect schemes; algorithms and methods; performance evaluation
- Application examples in distributed sensing and action

Keywords

Artificial intelligence, swarm intelligence, distributed robotics, sensor networks, modeling, machine-learning, control

Learning Prerequisites

Required courses

Fundamentals in analysis, probability, and programming for both compiled and interpreted languages

Recommended courses

Basic knowledge in statistics, specific programming language used in the course (C and Matlab), and signals and systems

Learning Outcomes

By the end of the course, the student must be able to:
- Design a reactive control algorithm

- Formulate a model at different level of abstraction for a distributed intelligent system
- Analyze a model of a distributed intelligent system
- Analyze a distributed coordination strategy/algorithm
- Design a distributed coordination strategy/algorithm
- Implement code for single robot and multi-robot systems
- Carry out systematic performance evaluation of a distributed intelligent system
- Apply modeling and design methods to specific problems requiring distributed sensing and action
- Optimize a controller or a set of possibly coordinated controllers using model-based or data-driven methods

Transversal skills

- Use both general and domain specific IT resources and tools
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate a capacity for creativity.
- Access and evaluate appropriate sources of information.
- Collect data.

Teaching methods

Ex-cathedra lecture and assisted exercises

Expected student activities

Attending lectures, carrying out exercises, and reading hand outs.

Assessment methods

Continuous control with final written exam.

Supervision

Office hours   Yes
Assistants   Yes
Forum   No

Resources

Bibliography

Lecture notes, selected papers and book chapters distributed at each lecture.

Websites

- http://disal.epfl.ch/teaching/distributed_intelligent_systems/

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=6391

Prerequisite for

R&D activities in engineering

COM-502   **Dynamical system theory for engineers**
Thiran Patrick

Cursus	Sem.	Type
Bioingénierie	MA1, MA3	Opt.
Informatique	MA1, MA3	Opt.
Mineur en Biocomputing	H	Opt.
Mineur en Neurosciences computationnelles	H	Opt.
Mineur en Systems Engineering	H	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Sciences du vivant	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Linear and nonlinear dynamical systems are found in all fields of science and engineering. After a short review of linear system theory, the class will explain and develop the main tools for the qualitative analysis of nonlinear systems, both in discrete-time and continuous-time.

Content

- **Introduction:** Dynamics of linear and non linear systems. Definitions; Unicity of a solution; Limit Sets, Attractors.
- **Linear Systems:** Solutions; Stability of autonomous systems, Geometrical analysis, connection with frequency domain analysis.
- **Nonlinear Systems:** Solutions; Examples. Large-scale notions of stability (Lyapunov functions). Hamltionian systems, gradient systems. Small-scale notions of stability (Linearization; stability and basin of attraction of an equilibrium point, stability of periodic solutions, Floquet Multipliers). Graphical methods for the analysis of low-dimensional systems. Introduction to structural stability, Bifurcation theory. Introduction to chaotic systems (Lyapunov exponents).
- The class is methodology-driven. It may present some limited examples of applications, but it is not application-driven.

Keywords

Dynamical Systems, Attractors, Equilibrium point, Limit Cycles, Stability, Lyapunov Functions, Bifurcations, Lyapunov exponents.

Learning Prerequisites

Required courses

- Linear algebra (MATH 111 or equivalent).
- Analysis I, II, III (MATH 101, 106, 203 or equivalent).
- Circuits & Systems II (EE 205 or equivalent) or a Systems & Signals class (MICRO 310/311 or equivalent).

Recommended courses

- A first-year Probabilty class, such as MATH-232, MATH-231, MATH-234(b), MATH-234(c), or equivalent.
- Analysis IV (MATH 207 or equivalent)

Important concepts to start the course

- Linear Algebra (vector spaces, matrix operations, including inversion and eigendecomposition).
- Calculus (linear ordinary differential equations; Fourier, Laplace and z-Transforms).
- Basic notions of topology.
- Basic notions of probability.

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze a linear or nonlinear dynamical system.
- Anticipate the asymptotic behavior of a dynamical system.
- Assess / Evaluate the stability of a dynamical system.
- Identify the type of solutions of a dynamical sytem.

Teaching methods

- Lectures (blackboard), 2h per week
- Exercise session, 1h per week.

Expected student activities

Exercises in class and at home (paper and pencil, and Matlab)

Assessment methods

1. Mid-term 20%
2. Final exam 80%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Course notes; textbooks given as reference on the moodle page of the course.

Notes/Handbook

Course notes, exercises and solutions provided on the moodle page of the course.

Websites

- http://moodle.epfl.ch/course/view.php?id=303

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=303

Prerequisite for

Any class using dynamical systems.

CS-473 **Embedded systems**

Beuchat René

Cursus	Sem.	Type
Computer engineering minor	H	Opt.
Génie électrique et électronique	MA1, MA3	Opt.
Informatique	MA1, MA3	Opt.
Mineur STAS Chine	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Oral
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

The main topics of this course are understanding and designing embedded system on a programmable circuit (FPGA). Students will be able to design a camera or a LCD controller on an FPGA in VHDL and will use their controller through a softcore processor.

Content

- Microcontrollers and their associated programmable interfaces (GPIO, Timer, SPI, A/D, PWM, interrupts)
- Hardcore/softcore processors (ie. NIOS II, ARM)
- Memory organizations, little/big endian
- Synchronous busses, dynamic bus sizing (ie. Avalon Bus in Memory Mapped mode)
- Processor busses, bussesrealized in a FPGA
- Serial busses(ie. UART, SPI, i2c, ...)
- How an LCD graphical screen and a CMOS camera work
- FPGA-based conception of Embedded Systems
- Embedded systems with processors on FPGAs

Laboratories provide knowledge & practice to develop an embedded system based on an FPGA device.

Keywords

microprocessors, microcontroller, FPGA, embedded systems, SoC, programmable interface

Learning Prerequisites

Required courses

Introduction to computing systems, Logic systems, Computer architecture

Recommended courses

Electronic, Programming (C/C++), Project System On Chip

Important concepts to start the course

- Computer architecture (processor, memory, programmable interfaces)
- Processor Architecture (PC, registers, ALU, instruction decoding, instruction execution)
- Knowledge of C programming language
- Knowledge of VHDL

Learning Outcomes

By the end of the course, the student must be able to:

- Design an embedded system on an FPGA
- Analyze a specific problem to be solved and propose an FPGA-based system to solve it
- Implement a solution to the given problem
- Realize and simulate the design
- Test the developed solution on an FPGA
- Use complex development tools and hardware debugging tools such as a logic analyzer and an oscilloscope

Transversal skills

- Use a work methodology appropriate to the task.
- Negotiate effectively within the group.
- Set objectives and design an action plan to reach those objectives.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Use both general and domain specific IT resources and tools
- Make an oral presentation.

Teaching methods

Ex cathedra and exercises, laboratories by specific sub-topics, final mini-project

Expected student activities

- Reading and deepening of course concepts
- Preparation of exercises performed in the laboratory
- Writing reports on different labs
- Realization of a final mini-project by group with oral presentation, report and demonstration

Assessment methods

With continuous control.
all labs 30%, mini-projet 20%, oral exam 50%

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Course on Moodle with forum

Resources

Virtual desktop infrastructure (VDI)
No

Bibliography
Teaching notes and suggested reading material on moodle
Specialized datasheets (micro-controllers, FPGA) and standards(ie, SPI, i2c, Amba, Avalon, etc)

Notes/Handbook
Documents and slides provided on moodle

Websites

- http://fpga4u.epfl.ch

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=1231

Prerequisite for

CS-476 Real-time embedded systems

CS-491 **Enterprise and service-oriented architecture**
Wegmann Alain

Cursus	Sem.	Type
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	6 weekly

Summary

In this course, we teach how to define the requirements for an IT system that would best serve the needs of an organisation. The course is taught using a non-conventional style in which the students learn mostly through the stress of a series of concrete experiences that mimic real-life situations.

Content

The goal of this course is closely related to IT, but a substantial part the material is related to business, as well as philosophy and psychology. Some formal models and programming are also taught, but the course can be taken by non IT students.
The exam might be written exam (to be agreed with the students at the beginning of the semester).
Detailed contents:
1) Business Part (4 weeks): practical experimentation and theoretical understanding of the key business processes of a manufacturing company : request for quotation process, development, planning, quality management and accounting.
2) Business / IT Part (6 weeks): specification of an IT system that provides after-sales service. We teach the following techniques : interviews, root cause analysis, analysis/design of the business services and of the IT services. The underlying theory is system thinking (Weinberg, Vickers) and the ISO/IEC standard RM-ODP.
3) IT Part (2 weeks): implementation - using BPMN visual programming - of an IT system prototype. Overview of the technological aspects of service-oriented architecture (wsdl, bpel, and soap protocols; rest architecture style).
4) Enterprise Architecture & Conclusions (2 weeks): Overview of the enterprise architecture frameworks (Zachman, TOGAF, Urba-EA). Synthesis and key learning points of the course.

Keywords

Request for quotation (RFQ), quotation, purchase order, leadtime, bill of material, development process, V process, spirale process, manufacturing planning, quality system, traceability, ISO 9000, financial statements, year-end book closing, ERP,
interview, contextual inquiry, root-cause analysis, ITIL, business service, IT service, requirements engineeing, SEAM system modeling, SEAM goal-belief modeling, SEAM behavior modeling, Vickers appreciative system, behavioral refinment, information modeling,
service-oriented architecture (SOA), BPMN, BPEL, WSDL, SOAP, REST.
enterprise architecture (EA), Zachman, TOGAF, Urba-EA.
Systemic paradigm, epistemology, ontology, axiology (ethics and esthetics).

Learning Outcomes

By the end of the course, the student must be able to:

- Describe business processes (sales, engineering, manufacturing, accounting)
- Assess / Evaluate business processes using ISO9000
- Coordinate business operations (role play)
- Analyze business needs for an IT system design
- Assess / Evaluate the IT processes using ITIL
- Conduct interviews with business stakeholders

- Formalize business requirements for an IT system design
- Design BPMN / BPEL workflow

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Use both general and domain specific IT resources and tools
- Write a scientific or technical report.
- Collect data.
- Make an oral presentation.
- Summarize an article or a technical report.

Teaching methods

Problem-based teaching

Resources

Bibliography

Beyer, H. and K. Holtzblatt (1999). "Contextual design." interactions **6**(1): 32-42.
http://dl.acm.org/citation.cfm?id=291229

Beyer, H. R. and K. Holtzblatt (1995). "Apprenticing with the customer." Commun. ACM **38**(5): 45-52.
http://dl.acm.org/citation.cfm?id=203365

Carr, N. G. (2003). "IT Doesn't matter", Harvard Business Review
https://hbr.org/2003/05/it-doesnt-matter

OMG (2004), Introduction to BPMN
http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf

Regev, G., H. Olivier, et al. (2011). Service Systems and Value Modeling from an Appreciative System Perspective. Second International Conference on Exploring Services Sciences. Geneva Switzerland, Springer-Verlag New York, Ms Ingrid Cunningham, 175 Fifth Ave, New York, Ny 10010 Usa. **82:** 146-157.
http://infoscience.epfl.ch/record/163961

Regev, G. and A. Wegmann (2004). Defining Early IT System Requirements with Regulation Principles: The Lightswitch Approach. Proceedings of the 12th IEEE International Requirements Engineering Conference (REí04). Kyoto, Japan**:** 144-153.
http://infoscience.epfl.ch/record/112299

Regev, G. and A. Wegmann (2005). Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering. Proceedings of the 13th IEEE International Conference on Requirements Engineering, IEEE Computer Society**:** 253-362.
http://infoscience.epfl.ch/record/112298

Rychkova, I., G. Regev, et al. Declarative Specification and Alignment Verification of Services in ITIL. First International Workshop on Dynamic and Declarative Business Processes (DDBP 2008). Munich, Germany.
http://infoscience.epfl.ch/record/129324

ITSMF (2007). An Introductory Overeiew of ITIL v3
http://www.best-management-practice.com/gempdf/itSMF_An_Introductory_Overview_of_ITIL_V3.pdf

Wegmann, A. (2003). On the Systemic Enterprise Architecture Methodology (SEAM)**:** 483-490.
http://infoscience.epfl.ch/record/89690

Wegmann, A., A. Kotsalainen, et al. (2008). Augmenting the Zachman Enterprise Architecture Framework with a Systemic Conceptualization. Proceedings of the 2008 12th International IEEE Enterprise Distributed Object Computing Conference, IEEE Computer Society: 3-13.
http://infoscience.epfl.ch/record/126293

Zachman, J. A. (1987). "A framework for information systems architecture." IBM Syst. J. **26**(3): 276-292.
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5387107

Tools:
Alloy **http://alloy.mit.edu/alloy/**

Intalio **http://ww.intalio.com/**

SeamCAD **htt://lams.epfl.ch/seamcad/**

Ressources en bibliothèque

- SeamCAD
- Intalio
- Declarative Specification and Alignment Verification of Services in ITIL / Rychkova
- Service Systems and Value Modeling from an Appreciative System Perspective / Regev
- Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering / Regev
- Contextual design / Beyer
- Quality Management Systems / ISO
- Introduction to BPMN / White
- An Introductory Overeiw of ITIL v3 / ITSMF
- On the Systemic Enterprise Architecture Methodology / Wegmann
- Defining Early IT System Requirements with Regulation Principles / Regev
- A Langage and Tool for relational models
- Augmenting the Zachman Enterprise Architecture Framework with a Systemic Conceptualization / Wegmann
- A framework for information systems architecture / Zachman
- Alloy

CS-452 **Foundations of software**

Odersky Martin

Cursus	Sem.	Type
Informatique	MA1, MA3	Obl.
Mineur en Informatique	H	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

The course introduces the foundations on which programs and programming languages are built. It introduces syntax, types and semantics as building blocks that together define the properties of a program part or a language. Students will learn how to apply these concepts in their reasoning.

Content

- simple types, lambda-calculus
- normalization, references, exceptions
- subtyping
- recursive types
- polymorphism
- advances features of the Scala type system

Learning Prerequisites

Recommended courses

Advanced topics in programming, Compiler construction

Important concepts to start the course

Functional programming
Basic knowledge of formal languages

Learning Outcomes

By the end of the course, the student must be able to:

- Argue design decisions of programming languages
- Assess / Evaluate soundness of type systems
- Compose higher-order functions
- Verify progress and preservation in type systems
- Work out / Determine operational equivalences
- Carry out projects of 2-3 weeks duration
- Distinguish valid from invalid proofs
- Implement type systems and operational semantics

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.

• Evaluate one's own performance in the team, receive and respond appropriately to feedback.

• Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.

• Manage priorities.

Teaching methods

Ex cathedra, practical exercises

Assessment methods

With continuous control

Resources

Ressources en bibliothèque

• Types and Programming Languages / Pierce

Websites

• http://lampwww.epfl.ch/teaching/index.html.en

MATH-483   **Gödel and recursivity**

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
Mathématiques pour l'enseignement	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Remarque

Pas donné en 2017-18 - Cours donnés en alternance tous les deux ans

Summary

Gödel incompleteness theorems and mathematical foundations of computer science

Content

Gödel's theorems:
Peano and Robinson Arithmetics. Representable functions. Arithmetic of syntax. Incompleteness, and undecidability theorems.

Recursivity :
Turing Machines and variants. The Church-Turing Thesis. Universal Turing Machine. Undecidable problems (the halting and the Post-Correspondance problems). Reducibility. The arithmetical hierarchy. Relations to Turing machines. Turing degrees.

Keywords

Gödel, incompleteness theorems, Peano arithmetic, Robinson arithmetic, decidability, recursively enumarable, arithmetical hierarchy, Turing machine, Turing degrees, jump operator, primitive recursive functions, recursive functions, automata, pushdown automata, regular languages, context-free languages, recursive languages, halting problem, universal Turing machine, Church thesis.

Learning Prerequisites

Recommended courses

Mathematical logic (or equivalent). A good understanding of 1st order logic is required - in particular the relation between syntax and semantics.

Important concepts to start the course

1st order logic: syntax, semantics, proof theory, completeness theorem, compactness theorem, Löwenheim-Skolem theorem.

Learning Outcomes

By the end of the course, the student must be able to:
- Estimate whether a given theory, function, language is recursive or no
- Decide the class that a language belongs to (regular, context-free, recursive,...)
- Elaborate an automaton
- Design a Turing machine

- Formalize a proof in Peano arithmetic
- Sketch the incompleteness theorems
- Propose a non-standard model
- Argue why Hilbert program failed

Teaching methods

Ex cathedra lecture and exercises

Assessment methods

Written: 3 hours

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Set Theory:

- Thomas Jech: Set theory, Springer 2006
- Kenneth Kunen: Set theory, Springer, 1983
- Jean-Louis Krivine: Theory des ensembles, 2007
- Patrick Dehornoy: Logique et théorie des ensembles; Notes de cours, FIMFA ENS: http://www.math.unicaen.fr/~dehornoy/surveys.html
- Yiannis Moschovakis: Notes on set theory, Springer 2006
- Karel Hrbacek and Thomas Jech: Introduction to Set theory, (3d edition), 1999

Recursion Theory :

- Micheal Sipser: Introduction to the Theory of Computation, Thomson Course Technology Boston, 2006
- Piergiorgio Odifreddi: Classical recursion theory, vol. 1 and 2, Springer, 1999
- Robert I. Soare: Recursively Enumerable Sets and Degres, A Study of Computable Functions and Computably Generated Sets, Springer-Verlag 1987
- Nigel Cutland: Computability, an introduction to recursive function theory, 1980
- Raymond M. Smullyan: recursion theory for methamathematics, Oxford, 1993

Proof theory :

- Wolfram Pohlers: Proof Theory, the first step into impredicativity, Springer, 2008
- A. S. Troelstra, H. Schwichtenberg, and Anne S. Troelstra: Basic proof theory, Cambridge, 2000
- S.R. Buss: Handbook of proof theory, Springer, 1998

Gödel's results :

- Raymond M. Smullyan: Gödel's incompleteness theorems, Oxford, 1992
- Peter Smith: An introduction to Gödel's theorems, Cambridge, 2008
- Torkel Franzen: Inexhaustibility, a non exhaustive treatment, AK Peteres, 2002
- Melvin Fitting: Incompleteness in the land of sets, King's College, 2007
- Torkel Franzen: Gödel's theorem: an incomplete guide to its use and abuse, AK Peters, 2005

Ressources en bibliothèque

- Théorie des ensembles / Krivine
- Introduction to Set theory / Hrbacek
- Proof Theory / Pohlers
- Notes on theory / Moschovakis
- Basic proof theory / Troelstra
- Introduction to the Theory of Computation / Sipser
- Handbook of proof theory / Buss
- Set theory / Jech
- Classical recursion theory / Odifreddi
- Recursion theory for methamathematics / Smullyan
- Set theory / Kunen
- Incompleteness in the land of sets / Fitting
- Recursively Enumerable Sets and Degres / Soare
- Gödel's theorem / Franzen
- Computability, an introduction to recursive function theory / Cutland
- Logique et théorie des ensembles / Dehornoy
- Gödel's incompleteness theorems / Smullyan
- An introduction to Gödel's theorems / Smith
- Inexhaustibility, a non exhaustive treatment / Franzen

Websites

- http://www.hec.unil.ch/logique/enseignement/recursivity

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14569

CS-486

Human computer interaction
Pu Faltings Pearl

Cursus	Sem.	Type
Data Science	MA2	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
Mineur STAS Chine	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly

Summary

This course starts with a simple premise: if a piece of software is useful, joyful and easy to use, people will want it. We thus teach methods for engaging user experience design. The course is limited to 30 students.

Content

Basic concepts of human-computer interaction
Introduction to HCI: its aims and goals
Design thinking
Qualitative research
User modeling: persona and empathy diagram
Task analysis
Visual design
Basic concepts of cognitive science
How people reason and mental models
How people learn to use software products
How people perceive the world
How people process information
Prototyping methods for HCI design
Storyboarding
Wireframe prototyping
Interactive prototyping
Video prototyping
Evaluation techniques
Cognitive walkthrough
Heuristic evaluation

Keywords

User experience design, design thinking, usability, design for engaging users, rapid prototyping techniques, evaluation with users, design challenge

Learning Prerequisites

Required courses
Introduction to Visual Computing

Recommended courses

Open to students enrolled in the Master and PhD programs in IC.

Important concepts to start the course
Design software for joyful user experience

Learning Outcomes

- Interview users and elicit their needs using the goal-directed design method
- Design and implement interfaces and intearctions
- Project management: set objectives and device a plan to achieve them
- Group work skills: discuss and identify roles, and assume those roles including leadership
- Communication: writing and argumentation skills

Teaching methods

Lectures, hands-on practice, design review

Expected student activities

Reading, case studies, peer discussions

Assessment methods

Individual project, group project, presentation

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)
No

Bibliography
About Face 3: The Essentials of Interaction Design by Alan Cooper et al. (available as e-book at NEBIS)
100 Things Every Designer Needs to Know about People by Susan Weinschenk (avaiable as e-book at NEBiS)

Ressources en bibliothèque

- About Face 3 / Cooper
- 100 Things Every Designer Needs to Know about People / Weinschenk

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=12291

EE-550

Image and video processing

Ebrahimi Touradj

Cursus	Sem.	Type
Génie électrique et électronique	MA1, MA3	Opt.
Humanités digitales	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Oral
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Project	2 weekly

Summary

This course covers fundamental notions in image and video processing, as well as covers most popular tools used, such as edge detection, motion estimation, segmentation, and compression. It is composed of lectures, laboratory sessions, and mini-projects.

Content

Introduction, acquisition, restitution
Two-dimensionnal signals and systems, Elementary signals, Properties of two-dimential Fourier transform, Discretization (spatial and spatio-temporal artefacts), Two-dimensional digital filters, Two-dimensional z-transform, Transfer function. Captors, monitors, printers, half-toning, color spaces.

Multi-dimensional filtres
Design of Infinite Impulse Response and Finite Impulse Response filters, Implementation of multi-dimensional filters, Directional decomposition and directional filters, M-D Sub-band filters, M-D Wavelets.

Visual perception
Neural system, Eye, Retina, Visual cortex, Model of visual system, Special effects, Mach phenomena and lateral inhibition, Color, Temporal vision.

Contour and feature extraction, segmentation
Local methods, Region based methods, Global methods, Canny, Mathematical morphology. Segmentation, Motion estimation

Visual information coding
Overview of the information theory and basics of rate-distortion, Conventional techniques : predictive coding, transform coding, subband coding, vector quantization, Advanced methods : multiresolution coding, perception based coding, region based coding, directional coding, fractals, Video coding : motion compensation, digital TV, High definition TV. Standards: JPEG, MPEG, H.261, H.263

Keywords

Contour detection, motion estimation, segmentation, human visual system, image compression, video compression

Learning Prerequisites

Required courses
Fundamental notions of signal processing

Recommended courses
Signal processing for communication

Important concepts to start the course
Sampling, quantization, transforms, programming, algorithms, systems

Learning Outcomes

By the end of the course, the student must be able to:
- Create simple image processing systems
- Create simple video processing systems
- Compare image processing tools
- Compare video processing tools
- Select appropriately optimal image and video processing tools

Transversal skills

- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Ex cathedra, laboratory sessions, mini-projects

Expected student activities

Written report of laboratory sessions, oral presentation of mini-projects, comprehension of various notions presented during the course, resolve simple problems of image and video processing.

Assessment methods

Laboratories, mini-project, oral exam

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Students are encouraged to ask for appoitment with the professor any time outside of teaching hours

Resources

Bibliography

handouts of image and video processing course
Fundamentals of Digital Image Processing, A. K. Jain

Ressources en bibliothèque

- Fundamentals of Digital Image Processing / Jain

Moodle Link

- http://moodle.epfl.ch/enrol/index.php?id=333

Prerequisite for

Semester projects , master thesis projects, doctoral thesis

MICRO-511 **Image processing I**

Unser Michaël, Van De Ville Dimitri Nestor Alice

Cursus	Sem.	Type
Bioingénierie	MA1, MA3	Opt.
Humanités digitales	MA1	Opt.
Informatique	MA1, MA3	Opt.
Microtechnique	MA1, MA3	Opt.
Mineur en Biocomputing	H	Opt.
Mineur en Neuroprosthétiques	H	Opt.
Mineur en Neurosciences computationnelles	H	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Sciences du vivant	MA1, MA3	Opt.
Sciences et ingénierie de l'environnement	MA1, MA3	Obl.

Language	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly

Summary

Introduction to the basic techniques of image processing. Introduction to the development of image-processing software and to prototyping in JAVA. Application to real-world examples in industrial vision and biomedical imaging.

Content

- Introduction. Image processing versus image analysis. Applications. System components.
- Characterization of continuous images. Image classes. 2D Fourier transform. Shift-invariant systems.
- Image acquisition. Sampling theory. Acquisition systems. Histogram and simple statistics. Linear and Max-Lloyd Quantization.
- Characterization of discrete images and linear filtering. z-transform. Convolution. Separability. FIR and IIR filters.
- Image-processing operations. Point operators (thresholding, histogram modification). Spatial operators (smoothing, enhancement, nonlinear filtering). Morphological operators.
- Introduction to image analysis and computer vision. Segmentation, edge detection, objet detection, image comparison.

Learning Prerequisites

Required courses

Signals and Systems I & II (or equivalent)

Important concepts to start the course

1-D signal processing: convolution, Fourier transform, z-transform

Learning Outcomes

By the end of the course, the student must be able to:
- Exploit the multidimensional Fourier transform
- Select appropriately Hilbert spaces and inner-products
- Optimize 2-D sampling to avoid aliasing

- Formalize convolution and optical systems
- Design digital filters in 2-D
- Analyze multidimensional linear shift-invariant systems
- Apply image-analysis techniques
- Construct image-processing software
- Elaborate morphological filters

Transversal skills

- Use a work methodology appropriate to the task.
- Manage priorities.
- Use both general and domain specific IT resources and tools

MICRO-512 — **Image processing II**

Unser Michaël, Van De Ville Dimitri Nestor Alice

Cursus	Sem.	Type
Bioingénierie	MA2, MA4	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
Microtechnique	MA2, MA4	Opt.
Mineur en Biocomputing	E	Opt.
Mineur en Neuroprosthétiques	E	Opt.
Mineur en Neurosciences computationnelles	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Science et ing. computationelles	MA2	Opt.
Sciences du vivant	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly

Summary

Study of advanced image processing; mathematical imaging. Development of image-processing software and prototyping in JAVA; application to real-world examples in industrial vision and biomedical imaging.

Content

- **Review of fundamental notions.** Multi-dimensional Fourier transform. Convolution. z-transform. Digital filters.
- **Continuous representation of discrete data.** Splines. Interpolation. Geometric transformations. Multi-scale decomposition (pyramids and wavelets).
- **Image transforms.** Karhunen-Loève transform (KLT). Discrete cosine transform (DCT). JPEG coding. Image pyramids. Wavelet decomposition.
- **Reconstruction from projections.** X-ray scanners. Radon transform. Central slice theorem. Filtered backprojection. Iterative methods.
- **Deconvolution.** Inverse and Wiener filtering. Matrix formulations. Iterative techniques (ART).
- **Statistical pattern classification.** Decision making. Bayesian classification. Parameter estimation. Supervised learning. Clustering.
- **Image analysis.** Pixel classification. Contour extraction and representation. Shape. Texture. Snakes and active contours.

Learning Prerequisites

Required courses

Image Processing I

Recommended courses

Signals and Systems I & II, linear algebra, analysis

Important concepts to start the course

Basic image processing and related analytical tools (Fourier transform, z-tranform, etc.)

Learning Outcomes

By the end of the course, the student must be able to:

- Construct interpolation models and continuous-discrete representations
- Analyze image transforms
- Design image-reconstruction algorithms
- Formalize multiresolution representations using wavelets
- Design deconvolution algorithms
- Perform image analysis and feature extraction
- Design image-processing software (plugins)
- Synthesize steerable filters

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Manage priorities.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

CS-487   **Industrial automation**

Pignolet-Oswald Yvonne Anne, Tournier Jean-Charles

Cursus	Sem.	Type
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	1 weekly

Remarque

This course can be taken by students of all engineering sections.

Summary

This course consists of two parts: 1) architecture of control systems, hands-on lab 2) handling of faults and failures in real-time systems, including fault-tolerant computing

Content

1. Processes and plants, control system architecture
2. Instrumentation, Programmable Logic Controllers and embedded computers
3. Industrial communication networks, field busses
4. Field device access protocols and application program interfaces
5. Human interface and supervision
6. Manufacturing Execution Systems
8. Real-time response and performance analysis
9. Dependability (Reliability, Availability, Safety, ...)

Keywords

Industrial Automation considers the control, communication and software in real-time systems: factories, energy production and distribution, vehicles and other embedded systems.
Industrial Automation encompasses the whole chain from sensors, motors, controllers, communication networks, operator visualization, archiving and up to manufacturing execution systems and enterprise resource management. It includes fault-tolerance against hardware and software faults and the evaluation methods.
This application-oriented course does not require previous knowledge in control theory. It complements communication systems courses with a focus on industrial application.

Learning Prerequisites

Recommended courses

Communication networks

Learning Outcomes

By the end of the course, the student must be able to:

- Characterize the (software) architecture of a control system
- Apply methods and trade-offs in real-time systems
- Analyze a plant

- Propose suitable automation solutions meeting the requirements
- Analyze the reliability, availability, safety of a system

Transversal skills

- Communicate effectively with professionals from other disciplines.
- Keep appropriate documentation for group meetings.
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

Oral presentation aided by slides, exercises as part of the lecture, practical work (workshop at Siemens and group assignment).

Expected student activities

- Understand material presented during lectures by asking questions and/or independent (online) searches
- Attend Siemens workshop (one full day on Siemens premises in Renens)
- Work on group assignment
- Hand-in report and slides for assignment on time

Assessment methods

Assignment 25% and final oral exam 75%

Resources

Bibliography

Nussbaumer, Informatique Industrielle (EPFL)
Olsson, Gustav & Rosen, Christian - industrial automation,Dept. of Industrial Electrical Engineering and Automation, Lund University, Lund, Sweden.

Ressources en bibliothèque

- Informatique Industrielle / Nussbaumer

Websites

- http://lamspeople.epfl.ch/kirrmann/

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14114

COM-402 **Information security and privacy**

Ford Bryan Alexander

Cursus	Sem.	Type
Data Science	MA2	Obl.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.
Science et ing. computationelles	MA2	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

This course will provide a broad overview of information security and privacy topics, with the primary goal of giving students the knowledge and tools they will need "in the field" in order to deal with the security/privacy challenges they are likely to encounter in today's "Big Data" world.

Content

- Data protection concepts: access control, encryption, compartmentalization
- Intrusion/hacking techniques, intrusion detection, advanced persistent threats
- Practices for management of personally identifying information
- Operational security practices and failures
- Data anonymization and de-anonymization techniques
- Information flow control
- Differential privacy
- Cryptographic tools for data security and privacy
- Policy, ethics, and legal considerations

Keywords

security, privacy, protection, intrusion, anonymization, cryptography

Learning Prerequisites

Required courses

Basic programming course or comparable demonstration of basic programming skills

Learning Outcomes

By the end of the course, the student must be able to:

- Understand the most important classes of information security/privacy risks in today's â##Big Dataâ## environment
- Exercise a basic, critical set of â##best practicesâ## for handling sensitive information
- Exercise competent operational security practices in their home and professional lives
- Understand at overview level the key technical tools available for security/privacy protection

Expected student activities

Attending lectures, solving assigned problems and "hands-on" exercises, reading and demonstrating understanding of provided materials.

COM-404 **Information theory and coding**

Telatar Emre

Cursus	Sem.	Type
Génie électrique et électronique	MA1, MA3	Opt.
Informatique et communications		Obl.
Informatique	MA1, MA3	Obl.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

The mathematical principles of communication that govern the compression and transmission of data and the design of efficient methods of doing so.

Content

1. Mathematical definition of information and the study of its properties.
2. Source coding: efficient representation of message sources.
3. Communication channels and their capacity.
4. Coding for reliable communication over noisy channels.
5. Multi-user communications: multi access and broadcast channels.
6. Lossy source coding : approximate representation of message sources.
7. Information Theory and statistics

Learning Outcomes

By the end of the course, the student must be able to:

• Formulate the fundamenal concepts of information theory such as entropy, mutual information, channel capacity

• Elaborate the principles of source coding and data transmission

• Analyze source codes and channel codes

• Apply information theoretic methods to novel settings

Teaching methods

Ex cathedra + exercises

Assessment methods

With continuous control

Resources

Ressources en bibliothèque

• Elements of Information Theory / Cover

Websites

• http://moodle.epfl.ch/enrol/index.php?id=14593

CS-430 — Intelligent agents

Faltings Boi

Cursus	Sem.	Type
Data Science	MA1	Opt.
Energie et durabilité	MA1, MA3	Opt.
Informatique et communications		Obl.
Informatique	MA1, MA3	Opt.
Ing. finance	MA1, MA3	Opt.
Mineur en Informatique	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	3 weekly

Summary

Software agents are widely used to control physical, economic and financial processes. The course presents practical methods for implementing software agents and multi-agent systems, supported by programming exercises, and the theoretical underpinnings including computational game theory.

Content

The course contains 4 main subject areas:

1) Basic models and algorithms for individual agents:
game-playing algorithms, reactive agents and reinforcement learning. Models and algorithms for rational, goal-oriented behavior in agents.
2) Multi-agent systems:
multi-agent planning, distributed algorithms for constraint satisfaction, coordination techniques for multi-agent systems.
3) Self-interested agents:
Models and algorithms for implementing self-interested agents motivated by economic principles: elements of computational game theory, models and algorithms for automated negotiation, social choice, mechanism design, electronic auctions and marketplaces.
4) Implementing multi-agent systems:
Agent platforms, ontologies and markup languages, web services and standards for their definition and indexing.

Learning Prerequisites

Recommended courses

Intelligence Artificielle or another introductory course to AI

Learning Outcomes

By the end of the course, the student must be able to:

- Choose and implement methods for rational decision making in software agents, based on decision processes and AI planning techniques
- Choose and implement methods for efficient rational decision making in teams of multiple software agents
- Model scenarios with multiple self-interested agents in the language of game theory
- Evaluate the feasibility of achieving goals with self-interested agents using game theory
- Design, choose and implement mechanisms for self-interested agents using game theory
- Implement systems of software agents using agent platforms

Teaching methods

Ex cathedra, practical programming exercises

Expected student activities

Lectures: 3 hours
Reading: 3 hours
Assignments/programming: 4 hours

Assessment methods

Mini-projects and exercises 40%, final exam 60%

Resources

Bibliography

Michael Wooldridge : An Introduction to MultiAgent Systems - Second Edition, John Wiley & Sons, 2009Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (2nd/3rd Edition), Prentice Hall Series in Artificial Intelligence, 2003/2009.

Ressources en bibliothèque

- Artificial Intelligence: A Modern Approach / Russell
- An Introduction to MultiAgent Systems / Wooldridge

Websites

- http://liawww.epfl.ch/
- http://moodle.epfl.ch/

CS-431   Introduction to natural language processing

Chappelier Jean-Cédric, Rajman Martin

Cursus	Sem.	Type
Data Science	MA2	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

The objective of this course is to present the main models, formalisms and algorithms necessary for the development of applications in the field of natural language information processing. The concepts introduced during the lectures will be applied during practical sessions.

Content

Several models and algorithms for automated textual data processing will be described: (1) morpho-lexical level: electronic lexica, spelling checkers, ...; (2) syntactic level: regular, context-free, stochastic grammars, parsing algorithms, ...; (3) semantic level: models and formalisms for the representation of meaning, ...
Several application domains will be presented: Linguistic engineering, Information Retrieval, Text mining (automated knowledge extraction), Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

Natural Language Processing; Computationnal Linguisitics; Part-of-Speech tagging; Parsing

Learning Outcomes

By the end of the course, the student must be able to:
- Compose key NLP elements to develop higher level processing chains
- Assess / Evaluate NLP based systems
- Choose appropriate solutions for solving typical NLP subproblems (tokenizing, tagging, parsing)
- Describe the typical problems and processing layers in NLP
- Analyze NLP problems to decompose them in adequate independant components

Teaching methods

Ex cathedra ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 25%, final exam 75%

Supervision

Office hours	No
Assistants	No

Forum   No

Resources

Virtual desktop infrastructure (VDI)
No

Bibliography

1. M. Rajman editor, "*Speech and Language Engineering*", EPFL Press, 2006.
2. Daniel Jurafsky and James H, Martin, "*Speech and Language Processing"*, Prentice Hall, 2008 (2nd edition)
3. Christopher D. Manning and Hinrich Schütze, "*Foundations of Statistical Natural Language Processing*", MIT Press, 2000
4. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008
5. Nitin Indurkhya and Fred J. Damerau editors, "*Handbook of Natural Language Processing*", CRC Press, 2010 (2nd edition)

Ressources en bibliothèque

- Handbook of Natural Language Processing / Indurkhya
- Introduction to Information Retrieval / Manning
- Speech and Language Processing / Jurafsky
- Speech and Language Engineering / Rajman
- Foundations of Statistical Natural Language Processing / Manning

Websites

- http://coling.epfl.ch

CS-433 **Machine learning**

Jaggi Martin, Urbanke Rüdiger

Cursus	Sem.	Type
Data Science	MA1	Obl.
Humanités digitales	MA1	Opt.
Informatique et communications		Obl.
Informatique	MA1, MA3	Obl.
Managmt, tech et entr.	MA1, MA3	Opt.
Mineur en Biocomputing	H	Opt.
Mineur en Informatique	H	Obl.
Mineur en Neurosciences computationnelles	H	Obl.
Mineur en Systèmes de communication	H	Opt.
SC master EPFL	MA1, MA3	Obl.
Science et ing. computationelles	MA1, MA3	Opt.
Sciences du vivant	MA1, MA3	Opt.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly

Summary

Machine learning and data analysis are becoming increasingly central in many sciences and applications. In this course, fundamental principles and methods of machine learning will be introduced, analyzed and practically implemented.

Content

1. *Basic regression and classification concepts and methods: Linear models, overfitting, linear regression, Ridge regression, logistic regression, and k-NN.*
2. *Fundamental concepts: cost-functions and optimization, cross-validation and bias-variance trade-off, curse of dimensionality.*
3. *Unsupervised learning: k-Means Clustering, Gaussian mixture models and the EM algorithm.*
4. *Dimensionality reduction: PCA and matrix factorization, word embeddings*
5. *Advanced methods: generalized linear models, SVMs and Kernel methods, Neural networks and deep learning*

Keywords

• *Machine learning, pattern recognition, deep learning, data mining, knowledge discovery, algorithms*

Learning Prerequisites

Required courses

• Analysis I, II, III
• Linear Algebra
• Probability and Statistics (MATH-232)
• Algorithms (CS-250)

Recommended courses

• *Introduction to differentiable optimization (MATH-265)*
• *Linear Models (MATH-341)*

Important concepts to start the course

• *Basic probability and statistics (conditional and joint distribution, independence, Bayes rule, random variables, expectation, mean, median, mode, central limit theorem)*

• *Basic linear algebra (matrix/vector multiplications, systems of linear equations, SVD)*

• *Multivariate calculus (derivative w.r.t. vector and matrix variables)*

• *Basic Programming Skills (labs will use Python)*

Learning Outcomes

By the end of the course, the student must be able to:

• Define the following basic machine learning problems: Regression, classification, clustering, dimensionality reduction, time-series

• Explain the main differences between them

• Implement algorithms for these machine learning models

• Optimize the main trade-offs such as overfitting, and computational cost vs accuracy

• Implement machine learning methods to real-world problems, and rigorously evaluate their performance using cross-validation. Experience common pitfalls and how to overcome them

• Explain and understand the fundamental theory presented for ML methods

Teaching methods

• Lectures

• Lab sessions

• Course Projects

Expected student activities

Students are expected to:

• attend lectures

• attend lab sessions and work on the weekly theory and coding exercises

• work on projects using the code developed during labs, in small groups

Assessment methods

• Written final exam

• Continuous control (Course projects)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Christopher Bishop, Pattern Recognition and Machine Learning
- Kevin Murphy, Machine Learning: A Probabilistic Perspective
- Shai Shalev-Shwartz, Shai Ben-David, Understanding Machine Learning
- Michael Nielsen, Neural Networks and Deep Learning
- (Jerome Friedman, Robert Tibshirani, Trevor Hastie, The elements of statistical learning : data mining, inference, and prediction)

Notes/Handbook

github.com/epfml/ML_course

Websites

- https://mlo.epfl.ch/page-146520.html

COM-516 **Markov chains and algorithmic applications**

Lévêque Olivier, Macris Nicolas

Cursus	Sem.	Type
Data Science	MA1	Opt.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Remarque

The same course was given in Spring 2015-2016 under the name "Random Walks".

Summary

The study of random walks finds many applications in computer science and communications. The goal of the course is to get familiar with the theory of random walks, and to get an overview of some applications of this theory to problems of interest in communications, computer and network science.

Content

Part 1: Markov chains (~6 weeks):
- basic properties: irreducibility, periodicity, recurrence/transience, stationary and limiting distributions,
- ergodic theorem: coupling method
- detailed balance
- convergence rate to the equilibrium, spectral gap, mixing times
- cutoff phenomenon
Part 2: Sampling (~6 weeks)
- classical methods, importance and rejection sampling
- Markov Chain Monte Carlo methods, Metropolis-Hastings algorithm, Glauber dynamics, Gibbs sampling
- applications: function minimization, coloring problem, satisfiability problems, Ising models
- coupling from the past and exact simulation

Keywords

random walks, stationarity, ergodic, convergence, spectral gap, mixing time, sampling, Markov chain Monte Carlo, coupling from the past

Learning Prerequisites

Required courses

Basic probability course
Basic linear algebra and calculus courses

Recommended courses

Stochastic Models for Communications (COM-300)

Important concepts to start the course

Good knowledge of probability and analysis.
Having been exposed to the theory of Markov chains.

Learning Outcomes

By the end of the course, the student must be able to:
- Analyze the behaviour of a random walk
- Assess / Evaluate the performance of an algorithm on a graph
- Implement efficiently various sampling methods

Teaching methods

ex-cathedra course

Expected student activities

active participation to exercise sessions and implementation of a sampling algorithm

Assessment methods

midterm, mini-project, written exam

Resources

Bibliography

Various references will be given to the students during the course, according to the topics discussed in class.

Notes/Handbook

Lecture notes will be provided

Websites

- http://ipgold.epfl.ch/~leveque/Markov_Chains/

Prerequisite for

This course is not so to speak a prerequisite for other courses, but could complement well the course COM-512 on Networks out of control, as well as other courses in statistics.

COM-514   **Mathematical foundations of signal processing**

Kolundzija Mihailo, Parhizkar Reza, Scholefield Adam James

Cursus	Sem.	Type
Data Science	MA1	Opt.
Informatique et communications		Obl.
Informatique	MA1, MA3	Opt.
Mineur en Systems Engineering	H	Opt.
Mineur en Systèmes de communication	H	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly

Summary

Signal processing tools are presented from an intuitive geometric point of view which is at the heart of all modern signal processing techniques. Student will develop the mathematical depth and rigor needed for the study of advanced topics in signal processing.

Content

From Euclid to Hilbert (vector spaces; Hilbert spaces; approximations, projections and decompositions; bases)
Sequences and Discrete-Time Systems (sequences; systems; discrete-time Fourier transform; z-transform; DFT; multirate sequences and systems; filterbanks)
Functions and Continuous-Time Systems (functions; systems; Fourier transform; Fourier series)
Sampling and Interpolation (sampling and interpolation with finite-dimensional vectors, sequences, functions and periodic functions)
Approximation and Compression (approximation by polynomials, splines, and series truncation)
Localization and Uncertainty (localization for functions, sequences and bases; local Fourier and wavelet bases; time, frequency and resolution in the real world)
Compressed Sensing (overview and definitions; reconstruction methods and applications)

Learning Prerequisites

Required courses

Circuits and Systems

Recommended courses

Signal processing for communications

Learning Outcomes

By the end of the course, the student must be able to:
- Master the right tools to tackle advanced signal and data processing problems
- Develop an intuitive understanding of signal processing through a geometrical approach
- Get to know the applications that are of interest today
- Learn about topics that are at the forefront of signal processing research

Teaching methods

Ex cathedra with exercises

Expected student activities

Attending lectures, completing exercises

Assessment methods

Homeworks 20%, midterm (written) 30%, final exam (written) 50%

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

M. Vetterli, J. Kovacevic and V. Goyal, *"Signal Processing: Foundations"*, Cambridge U. Press, 2014. Available in open access at http://www.fourierandwavelets.org

Ressources en bibliothèque

• Signal Processing: Foundations / Vetterli

Websites

• http://lcav.epfl.ch/SP_Foundations

Moodle Link

• http://moodle.epfl.ch/course/view.php?id=13431

COM-405   **Mobile networks**

Hubaux Jean-Pierre

Cursus	Sem.	Type
Génie électrique et électronique	MA2, MA4	Opt.
Information security minor	E	Opt.
Informatique	MA2	Opt.
Mineur STAS Chine	E	Opt.
Mineur en Systèmes de communication	E	Opt.
SC master EPFL	MA2, MA4	Obl.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

This course provides a detailed description of the organization and operating principles of mobile communication networks.

Content

Introduction to wireless networks
Organization of the MAC layer
Wireless Local Area Networks - WiFi
Cellular networks
Mobility at the network and transport layers
Security and privacy in mobile networks

Keywords

Communication networks, protocols, mobility

Learning Prerequisites

Required courses
Computer Networks

Recommended courses
Principles of Digital Communications
Network security

Important concepts to start the course
Operating principles of communication protocols and layer organization.

Learning Outcomes

By the end of the course, the student must be able to:
• Synthesize the way a mobile network operates
• Interpret the behavior of such networks
• Propose evolutions to existing protocols
• Identify weaknesses, bottlenecks and vulnerabilities

Teaching methods

Ex cathedra lectures

Weekly quizzes
Exercise sessions
Hands-on exercises

Expected student activities

Class participation, quizzes, homework, hands-on exercises

Assessment methods

Quizzes + final exam

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	The lecturer will be present at most of the exercise sessions.

Resources

Bibliography

Handouts, recommended books (see course URL)

Websites

- http://mobnet.epfl.ch/

COM-430 — **Modern digital communications: a hands-on approach**

Rimoldi Bixio

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
Mineur en Systèmes de communication	H	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Practical work	2 weekly

Remarque

.

Summary

This course complements the theoretical knowledge learned in PDC with more advanced topics such as OFDM, MIMO, LDPC encoding/decoding, GPS positioning. This knowledge is put into practice with hands-on exercises based on Matlab and on a software-defined radio platform.

Content

1. Software radio : key concepts.
2. Matlab implementation of the signal processing chain to the level of detail in Principles of Digital Communications.
3. Channel modeling, estimation, equalization.
4. Implementation of a basic wireless communication system using a software-defined radio testbed.
5. OFDM and MIMO : theory and implementation.
6. CDMA in the context of a GPS system.
7. Decoding of a GPS signal and positioning.
8. LDPC codes, efficient encoding, and decoding based on the message passing algorithm.

Keywords

Wireless, PFDM, Diversity, Waterfillingn, Coding, Multi-user communication, GPS, CDMA, LDPC, MMSE, software radio, equalization.

Learning Prerequisites

Required courses

COM-302 Principles of digital communication or equivalent

Important concepts to start the course

Solid understanding of linear algebra and probability as well as real and complex analysis.

Learning Outcomes

By the end of the course, the student must be able to:

• Design and implement and advanced digital communication system (data rate, spectral bandwidth, energy requirements, error probability, implementation complexity)

• Model physical properties of wired and wireless communication channels

- Implement various parts of a "physical-layer" digital communication system
- Understand what software-defined radio is all about

Teaching methods

Ex cathedra lectures and small projects

Expected student activities

Follow lectures; guided as well as independent work on projects

Assessment methods

Written and practical midterm and final exam during the semester.
40% midterm exam, 60% final exam.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Notes/Handbook

Lecture notes

COM-512

Networks out of control

Celis Laura Elisa, Thiran Patrick

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique	MA2	Opt.
Mineur en Systems Engineering	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Remarque

Cours biennal donné les années impaires (donné en 2017-18)

Summary

The goal of this class is to acquire mathematical tools and engineering insight about networks whose structure is random, as well as decentralized processes that take place on these networks.

Content

- Course Introduction: Tree Percolation, Branching Processes
- Random Graphs 1: Models, Threshold Functions, Appearance of Subgraphs
- Random Graphs 2: Giant Component and Connectivity
- Random Graphs 3: Other models: the Random Regular Graph, Small World Networks, Scale-Free Networks.
- Random Geometric Graphs: Introduction to Percolation Theory.
- Evolution, Dynamics and Inference 1: Epidemics, Network and Source Discovery.
- Evolution, Dynamics and Inference 2: Information Cascades.
- Evolution, Dynamics and Inference 3: Network Navigation and Price of Anarchy.
- Applications 1: Network Formation Games.
- Applications 2: Homophily, Structural Balance.

Keywords

Random graphs, percolation theory, social networks, communication networks.

Learning Prerequisites

Required courses

Stochastic models in communication (COM-300), or equivalent.

Important concepts to start the course

Basic probability and stastistics; Markov chains; basic combinatorics.

Learning Outcomes

By the end of the course, the student must be able to:
- Analyze social and communication systems

- Model such systems as stochastic models
- Compute key properties of these models

Teaching methods

Ex cathedra lectures, exercises, mini-project

Expected student activities

Attending lectures, bi-weekly homeworks, mini-project incl. student presentation at the end of semester, final exam.

Assessment methods

1. Homeworks 10%
2. Mini-project 40%
3. Final exam 50%.

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Bibliography

- A. D. Barbour, L. Holst and S. Janson, Poisson Approximation, Oxford Science Publications, 1992.
- B. Bollobas, Random Graphs (2nd edition), Cambridge University Press, 2001.
- R. Durrett, Random Graph Dynamics, Cambridge University Press, 2006 (electronic version).
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010 (electronic version).
- G. Grimmett, Percolation (2nd edition), Springer, 1999.
- S. Janson, T. Luczak, A. Rucinski, Random Graphs, Wiley, 2000.
- R. Meester and R. Roy, Continuum Percolation, Cambridge University Press, 1996.

Ressources en bibliothèque

- Random Graphs / Bollobas
- Random Graphs / Janson
- Continuum Percolation / Meester
- Random Graph Dynamics / Durrett
- Networks, Crowds and Markets / Easley
- Poisson Approximation / Barbour
- Percolation / Grimmett

Notes/Handbook

Class notes will be available on the course website.

Websites

- http://icawww1.epfl.ch/class-nooc/

MATH-489   **Number theory in cryptography**

Jetchev Dimitar Petkov

Cursus	Sem.	Type
Informatique	MA2	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Mathématiques pour l'enseignement	MA2, MA4	Obl.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

The goal of the course is to introduce basic notions from public key cryptography (PKC) as well as basic number-theoretic methods and algorithms for cryptanalysis of protocols and schemes based on PKC.

Content

Basic notions and algorithms from public key cryptography such as RSA, ElGamal, key exchange protocols, zero knowledge proofs. Main topics may include, but are not limited to

- modular and finite field arithmetic
- primality testing
- polynomial and integer factorization algorithms
- index calculus and discrete logarithm-based schemes
- elliptic curve cryptography
- basic notions from lattice-based cryptography

Keywords

public key cryptography, key exchange, digital signatures, zero knowledge proofs, RSA, ElGamal, integer factorization, index calculus, elliptic curve cryptography

Teaching methods

lecture notes, additional references

Assessment methods

Theoretical assignments: Weekly problem sets focusing on number-theoretic and complexity-theoretic aspects. Theoretical assignments will count for 30% of the final grade.

Programming assignments: All of the programming exercises will be in SAGE which is a Python-based computer algebra system. No prior experience with SAGE or Python is required. Programming assignments will count for 30% of the final grade.
One mid-term exam (15% of the final grade) and **one final exam** (25% of the final grade). Both exams will test theoretical understanding as well as understanding of the algorithms and protocols. The exams will include no SAGE programming exercises. If needed, algorithms could be presented with pseudo-code.

CS-439   **Optimization for machine learning**

Jaggi Martin

Cursus	Sem.	Type
Data Science	MA2	Obl.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.
Science et ing. computationelles	MA2	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.

Content

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.
Basic Contents:
Convexity, Gradient Methods, Proximal algorithms, Stochastic and Online Variants of mentioned methods, Coordinate Descent Methods, Subgradient Methods, Frank-Wolfe, Accelerated Methods, Primal-Dual context and certificates, Lagrange and Fenchel Duality, Second-Order Methods, Quasi-Newton Methods. Black-Box Optimization.
Advanced Contents:
Parallel and Distributed Optimization Algorithms, Synchronous and Asynchronous Communication.
Computational and Statistical Trade-Offs (Time vs Data vs Accuracy). Variance Reduced Methods, and Lower Bounds.
Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods, Alternating minimization for matrix and tensor factorizations

Keywords

Optimization, Machine learning

Learning Prerequisites

Recommended courses

• CS-433 Machine Learning

Important concepts to start the course

• Previous coursework in calculus, linear algebra, and probability is required.
• Familiarity with optimization and/or machine learning is useful.

Learning Outcomes

By the end of the course, the student must be able to:

• Assess / Evaluate the most important algorithms, function classes, and algorithm convergence guarantees
• Compose existing theoretical analysis with new aspects and algorithm variants.

- Formulate scalable and accurate implementations of the most important optimization algorithms for machine learning applications
- Characterize trade-offs between time, data and accuracy, for machine learning methods

Transversal skills

- Use both general and domain specific IT resources and tools
- Summarize an article or a technical report.

Teaching methods

- Lectures
- Exercises with Theory and Implementation Assignments

Expected student activities

Students are expected to:
- Attend the lectures and exercises
- Give a short scientific presentation about a research paper
- Read / watch the pertinent material
- Engage during the class, and discuss with other colleagues

Assessment methods

- Final Exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)
No

Websites

- http://coming soon

CS-596 **Optional project in computer science**
Profs divers *

Cursus	Sem.	Type
Informatique	MA1, MA2, MA3	Opt.
Mineur en Informatique	E, H	Opt.

Language	English
Credits	8
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**2 weekly**
Project	2 weekly

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Subject to be chosen among the themes proposed on the web site :

http://ic.epfl.ch/semester_projects_by_laboratory

Learning Outcomes

By the end of the course, the student must be able to:
• Organize a project
• Assess / Evaluate one's progress through the course of the project
• Present a project

Transversal skills

• Write a scientific or technical report.
• Write a literature review which assesses the state of the art.

Teaching methods

Individual and independant work, under the guidance of a professor or an assistant.

Assessment methods

Oral presentation and written report.

Resources

Virtual desktop infrastructure (VDI)
Yes

Websites

• http://ic.epfl.ch/semester_project

COM-503   **Performance evaluation**

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique et communications		Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Oral
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly

Remarque

cours biennal donné les années paires (pas donné en 2017-18)

Summary

In this course you will learn the methods and techniques that are used to perform a good performance evaluation during a research or development project.

Content

MethodologyA Performance Evaluation Methodology. The scientific method. Dijkstra and Occam's principle.
Statistics and Modeling.
Statistics and modeling, why and how. Comparing systems using sampled data. Regression models. Factorial analysis. Stochastic load and system models. Load forecasting. The Box-Jenkins method.
Practicals.
Using a statistics package (Matlab). Measurements. Discrete event simulation. Stationarity and Steady State. Analysis of simulation results. Perfect Simulations.
Elements of a Theory of Performance. Performance of systems with waiting times. Utilization versus waiting times. Operational laws. Little's formula. Forced flows.law. Stochastic modeling revisited. The importance of the viewpoint. Palm calculus. Application to Simulation Performance patterns in complex systems. Bottlenecks. Congestion phenomenon. Performance paradoxes.

Mini-Project proposed by student.

Learning Prerequisites

Required courses

A first course on probability
A first course on programming

Learning Outcomes

By the end of the course, the student must be able to:

- Estimate confidence intervals
- Design a simulation method
- Critique performance metrics and factors
- Organize a performance evaluation study
- Quantify performance
- Conduct a performance analysis
- Synthesize performance results

• Systematize factors and metrics

Transversal skills

- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking

Teaching methods

Lectures + labs + miniproject

Assessment methods

With continuous control

Resources

Bibliography

Performance Evaluation of Computer and Communication Systems, Le Boudec Jean-Yves, EPFL Press 2010 also available online at perfeval.epfl.ch

Ressources en bibliothèque

- Performance evaluation of computer and communication systems / Le Boudec

Websites

- http://perfeval.epfl.ch

CS-489 | **Personal interaction studio**

Huang Jeffrey

Cursus	Sem.	Type
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Project	4 weekly

Summary

As we move towards a design economy, the success of new products, systems and services depend increasingly on the excellence of personal experience. This course introduces students to the notion and practice of experience and interaction design following a hands-on, studio-based approach.

Content

STUDIO BRIEF : THE FUTURE OF MAPS
The map itself has emerged as one of the most ubiquitous interactive objects of our digital mobile age. It plays a significant role in our contemporary understanding of information (abstract data and physical spaces). This term, the studio aims to reinterpret the map as a digital, live, interactive artifact. The goal is to create meaningful interactive datadriven maps as both a digital visualization interface, as well as, in the form of a physically sited exhibition.
CONTENT
The course consists of a non-linear/iterative process of 'hackathon-like' and 'creative-coding' workflow. The course will contain a series of iterative design props – 'problem maps', 'value maps', 'data maps' and 'future maps' – as an apparatus to construct a network of understandings, and create meaningful user experiences for a final design proposal/product.
I. Problem Maps
1. Precedents survey and analysis
2. Intuitive approach to problematizing design
3. Generating opportunities for design interventions
II. Value Maps
1. Identifying and mapping values (economic/social/cultural) to design interventions
2. Analysis of identified value hierarchies with targeted market personas
III. Data Maps
1. Data source availability (proprietary/non-proprietary)
2. Data service availability
3. Data mining and logging methods
IV. Future Maps
1. Vision Proposal
2. Concept Design
3. Schematic Design of components/architectures
4. Rapid Prototyping with creative coding tools and workflows
5. Design Optioneering with evaluation and testing
6. Design Marketing with scenarios and storyboards
7. Scaling for different forms of realization (esp. physically sited exhibition)

Keywords

User Experience (UX) Design, Design Thinking, Creative Coding, Hackathon, Open Source, Optioneering, Iterative Prototyping

Learning Prerequisites

Required courses

Bachelor in Computer Science or equivalent

Learning Outcomes

By the end of the course, the student must be able to:

- Identify issues of experience design in relation to an actual design project
- Perform rigorous analysis of the problem space and map the design opportunities (problem seeking, value proposition and data inventories)
- Develop alternative design concepts for future artifacts (in 2016: live maps)
- Translate design concepts into meaningful experiences through iterative prototyping at appropriate scales and levels of granularity (creative coding)
- Create convincing arguments for the design propositions and persuasive visual and tangible evidence

Teaching methods

Hackathon, Creative coding, Lectures, Design reviews, Presentations, Group projects

Expected student activities

Hackathon, Group discussion, Case studies, Design Reviews, Pin-Up, Desk Crits

Assessment methods

Grading will be based upon the quality of the projects in the preliminary stages (10% problem maps, 10% value maps, 10% data maps), intermediary reviews (20% future maps) and in the final review (50%). Final projects will be reviewed and assessed based on their conceptual strength, the coherence of their translation into prototypes, their narrative clarity and experiential power, and the persuasiveness of their communication, both orally and through the presented artifacts.

Supervision

Office hours   Yes
Assistants   No

Resources

Bibliography

To be made available during the course

CS-522   **Principles of computer systems**

Argyraki Aikaterini, Candea George

Cursus	Sem.	Type
Informatique et communications		Obl.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	During the semester
Workload	210h
Weeks	14
Hours	**4 weekly**
Lecture	4 weekly

Summary

This advanced graduate course focuses on key design principles underlying successful computer and communication systems, and teaches how to solve real problems using ideas, techniques, and algorithms from operating systems, networks, databases, programming languages, and computer architecture.

Content

A modern computer system spans many layers: applications, libraries, operating systems, networks, and hardware devices. Building a good system entails making the right trade-offs (e.g., between performance, durability, and correctness) and understanding emergent behaviors - the difference between great system designers and average ones is that the really good ones make these trade-offs in a principled fashion, not by trial-and-error.

In this course we develop such a principled framework for system design, covering the following topics:

- Modularity, Abstraction, and Layering
- Indirection and Naming
- Locality
- End-to-end / State partitioning
- Virtualization
- Atomicity and Consistency
- Redundancy and Availability
- Interpretation, Simulation, Declarativity
- Laziness vs. Speculation
- CAP Theorem, DQ Principle, Harvest/Yield
- Least Privilege, Minimum TCB

Learning Prerequisites

Required courses

Principles of Computer Systems (POCS) is targeted at students who wish to acquire a deep understanding of computer system design or pursue research in systems. It is an intellectually challenging, fast paced course, in which mere survival requires a solid background in operating systems, databases, networking, programming languages, and computer architecture. The basic courses on these topics teach how the elemental parts of modern systems work - POCS picks up where the basic courses leave off and focuses on how the pieces come together to form useful, efficient systems. To do well in POCS, a student must master the material of the following courses:

- COM-208 Computer networks
- CS-208 Computer architecture
- CS-210 Functional programming

- CS-305 Software engineering
- CS-322 Introduction to database systems
- CS-323 Operating systems

Recommended courses

The following EPFL courses cover material that significantly help students' understanding of POCS concepts; however, these courses are not strictly required:

- CS-320: Computer language processing
- CS-470: Advanced computer architecture
- CS-422: Database systems
- COM-407: TCP/IP networking

Learning Outcomes

By the end of the course, the student must be able to:

- Design computer and communication systems that work well
- Make design trade-offs (e.g., performance vs. correctness, latency vs. availability)
- Anticipate emergent system behaviors (e.g., failure cascades, security vulnerabilities)
- Integrate multiple techniques, ideas, and algorithms from different fields of computing/communication into a working system

Teaching methods

- Online video lectures
- Ex cathedra
- Small-group discussions and exercises
- Projects

Expected student activities

- Complete assigned reading and writing assignments
- Assimilate online video lectures
- Attend recitations and plenary sessions
- Participate actively in class (physically and online)
- Work in a team on design projects

Assessment methods

Throughout semester

- 20% homework
- 40% design projects
- 40% exam during semester

Supervision

Office hours	Yes
Assistants	Yes

Forum   Yes
Others   See http://pocs.epfl.ch/

Resources

Bibliography
See http://pocs.epfl.ch for up-to-date bibliography.

Ressources en bibliothèque

- Principles of computer system design : an introduction / Saltzer

Websites

- http://pocs.epfl.ch

MATH-467   **Probabilistic method**

Cursus	Sem.	Type
Informatique	MA2	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Mathématiques pour l'enseignement	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Remarque

pas donné en 2017-18

Summary

We systematically explore the exciting fact that randomness (i.e., coin flipping) can be used profitably to construct various mathematical structures with unexpected and often paradoxical properties, and to efficiently solve otherwise hopelessly difficult computational tasks.

Content

- Linearity of expectation
- Applications in combinatorics and number theory
- Randomized algorithms (sorting, convex hull, linear programming)
- The second moment method
- Random graphs

Keywords

random variable, expected value, probabilistic method, random graph, coloring

Learning Prerequisites

Required courses

Probability theory

Recommended courses

Discrete Mathematics or Graph Theory

Important concepts to start the course

Graph, random variable, expectation, variance, binomial coefficients, asymptotics

Learning Outcomes

By the end of the course, the student must be able to:

- Define and explain basic concepts in probability and discrete mathematics
- Define threshold functions, and analyze their asymptotic behavior
- Prove explain, and apply the first and second moment methods
- Prove explain, and apply the Local Lemma

- Solve exercises, design randomized algorithms
- Describe and explain the evolution of random graphs

Transversal skills

- Summarize an article or a technical report.
- Demonstrate the capacity for critical thinking
- Assess progress against the plan, and adapt the plan as appropriate.

Teaching methods

Lectures and exercises

Expected student activities

Attending the lectures, solving the exercises, reading sections from the textbook

Assessment methods

Exam written

Resources

Bibliography

Noga Alon-Joel Spencer: The Probabilistic Method (Wiley)
Stasys Jukna: Extremal Combinatorics (Springer)

CS-498	**Projet en informatique II**
	Profs divers *

Cursus	**Sem.**	**Type**
Informatique	MA1, MA2, MA3	Obl.

Language	English
Credits	12
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	360h
Weeks	14
Hours	**2 weekly**
Project	2 weekly

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Subject to be chosen among the themes proposed on the web site :

http://ic.epfl.ch/semester_projects_by_laboratory

Learning Outcomes

By the end of the course, the student must be able to:
- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Transversal skills

- Write a scientific or technical report.
- Write a literature review which assesses the state of the art.

Assessment methods

Written report and oral presentation

Resources

Virtual desktop infrastructure (VDI)
No

Websites
- http://ic.epfl.ch/semester_project

CS-476

Real-time embedded systems

Beuchat René

Cursus	Sem.	Type
Computer engineering minor	E	Opt.
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	MA2	Opt.
Mineur STAS Chine	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Summary

A real time system is subject to important temporal constraints. This course is about understanding where processing time is spent and what a designer can do in order to achieve real-time processing systems. Some solutions are Multiprocessors, accelerators, custom instructions, specialized hardware.

Content

During this course, response time measurements of interrupts are studied in laboratories, such as for example: the influence of dynamic memories, cache memories, compilation flags. Interrupts response time measurements, task commutations and synchronizations primitives are carried out on an embedded system based on an FPGA.
The course includes the study of embedded systems management models through polling, interrupts and using a real time kernel with its task management and synchronization primitives.
Specialized programmable interfaces are implemented in VHDL to help with these measurements. A real time kernel is studied and used during the labs. An acquisition system is implemented and the gathered data is transmitted by a Web server. To ensure the real time acquisition and reading by the Web server, a multiprocessor system is developed and implemented on an FPGA.
An Accelerator designed in VHDL makes it possible to facilitate the optimization of functions through hardware on an FPGA. Cross development tools are used.
Each topic is treated by a theoretical course and an associated laboratory. The laboratories are realized on an FPGA board including a hardcore multiprocessor. A real time operating system is studied and used with the laboratories.

Keywords

Real Time, FPGA, SOC, microprocessor, hardware accelerator, custom instruction, Real Time OS

Learning Prerequisites

Required courses

Introduction to computing systems, Logic systems, Computer architecture

Recommended courses

Embedded Systems, Real time Programming

Important concepts to start the course

Programmable Logic Architecture (FPGA), Computer Architecture, VHDL, C programming, Real Times basic knowledge (semaphor, synchronization)

Learning Outcomes

By the end of the course, the student must be able to:

- Design a multiprocessor system on an FPGA
- Analyze the performance of a real time embedded system
- Use design tools for SOC conception on an FPGA
- Implement a complete real-time system based on a multiprocessor design on an FPGA
- Test the realized system
- Defend the choices during the design phases

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Communicate effectively, being understood, including across different languages and cultures.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Ex cathedra, laboratories and a mini-project

Expected student activities

- 3 groups of laboratories on specific topics, with a report by group for each of them, 3-4 weeks/topic;
- A final mini-project to practically synthesize the content of the course, with the design of a multiprocessor system on an FPGA, including for example a Web-server, a camera controller, a specific algorithm to be implemented in an FPGA hardware accelerator, 3~4 weeks for this mini-project

Assessment methods

Continuous control with reports and oral presentation
all labs 50% + final mini-project 50%

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)
No

Bibliography
Teaching notes and suggested reading material.
Specialized datasheets (ie.ex. FPGA et specific microcontrollers) and standards.

Notes/Handbook
Slides and documents on moodle

Websites

- http://fpga4u.epfl.ch

Moodle Link
http://moodle.epfl.ch/course/view.php?id=391

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

COM-413 **Real-time networks**

Decotignie Jean-Dominique

Cursus	Sem.	Type
Energie et durabilité	MA2, MA4	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**2 weekly**
Lecture	2 weekly

Summary

At course completion, the student will be able to analyse the real-time properties of a communication network; and will also be able to create a new solution either balancing the tradeoffs between the different design parameters or composing building blocks. Applications to multimedia, transports,

Content

1. Introduction (hierarchy in communications, motivation for networks, types of applications)
2. Requirements (delay, jitter, predictability, topology, cost, etc.)
3. Communication systems architecture and its influence on temporal behavior(OSI model, communication models, real-time paradigms : Time-Triggered vs. Event-Triggered, interworking)
4. Fieldbusses and how real-time performance assessment : FIP and CAN as examples
5. Ethernet, industrial Ethernet and real-time Ethernet
6. Wireless communications and their impact on real-time guarantees
7. IEEE 802.11 and IEEE 802.11e
8. Bluetooth, IEEE 802.15.4 (ZigBee) and wireless sensor networks
9. Real-time in wireless sensor networks

Keywords

real-time, networking, wireless, wireless sensor networks, medium access control, quality of service

Learning Prerequisites

Required courses
none

Recommended courses
real-time systems, protocols

Important concepts to start the course
Protocols and real-time system background

Learning Outcomes

By the end of the course, the student must be able to:

- master real-time techniques in wired and wireless networking
- modelling of quality of service requirements
- deep knowledge of real-time medium access control techniques
- exercize the real-time garantee evaluation techniques
- capability to design a new real-time solution

Transversal skills

- Communicate effectively, being understood, including across different languages and cultures.

Teaching methods

Ex cathedra + student presentations + exercises

Expected student activities

Learning the course material, reading, presentation and discussion of a scientific paper as an introduction to research

Assessment methods

Mid-term presentation 50% and final exam 50%

Supervision

Office hours	No
Assistants	No
Forum	Yes

Resources

Bibliography

See course URL

Websites

- http://lamspeople.epfl.ch/decotignie/
- http://moodle.epfl.ch

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=10761

EE-511   **Sensors in medical instrumentation**

Aminian Kamiar

Cursus	Sem.	Type
Bioingénierie	MA2, MA4	Opt.
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	MA2	Opt.
Microtechnique	MA2, MA4	Opt.
Mineur en Technologies biomédicales	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Sciences du vivant	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

Fundamental principles and methods used for physiological signal conditioning. Resistive, capacitive, inductive, piezoelectric and optical techniques used to detect and convert physiological information's to electrical signals. Laboratory and ambulatory devices for monitoring and therapy.

Content

1. **Physiological Mesurands**
Biopotentials; bioimpedance; mechanical, acoustic and thermal signals
2. **Noise in medical instrumentation**
Source and nature of the noise; noise reduction; instrumentation amplifier
for biopotential measurement
3. **Biopotential measurement**
Electrodes; ECG, EMG and EEG measurement
4. **Resistive sensors**
Thermistor and its biomedical applications; strain gage for the
measurement of blood pressure; force and accelerations of the body
5. **Inductive sensors**
Simple and mutual inductance and its medical applications
6. **Capacitive sensors**
Respiratory flow measurement by the gradient of pressure
7. **Piezoelectric sensors**
Force platform, accelerometer, angular rate sensor for the measurement
of tremors and body movements, ultrasound transducer : measurement of
pressure and flow rate
8. **Optical sensors**
Photoplethysmography; pulsed oxymetry
9. **Example of applications**

Keywords

sensors, instrumentation, biomedical devices, physiological measurement, monitoring

Learning Prerequisites

Required courses

courses en electrical circuit, basic electronics

Recommended courses

measuring systems or electronics or sensors

Important concepts to start the course

basic electronics, basic physics

Learning Outcomes

By the end of the course, the student must be able to:

- Choose techniques detecting and convert physiological information's to electrical signals
- Exploit fundamental principles and methods used for physiological signal conditioning
- Design measuring devices
- Interpret error, noise in biomedical measuring systems

Transversal skills

- Use a work methodology appropriate to the task.
- Communicate effectively with professionals from other disciplines.

Teaching methods

Ex cathedra, with exercises

Expected student activities

home work, short quizzes during semester

Assessment methods

Written

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Medical Instrumentation : Application and design, JG Webster

Ressources en bibliothèque

- Medical Instrumentation / Webster

Notes/Handbook

Slides copies (to be completed during the lectures)
Polycopies (in French only)

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=2571

Prerequisite for

Semester project and Master project

MATH-318   **Set theory**

Duparc Jacques

Cursus	Sem.	Type
Informatique	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
Mathématiques pour l'enseignement	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

Set Theory as a foundational system for mathematics. Relative consistency of the Axiom of Choice and the Continuum Hypothesis.

Content

Set Theory: ZFC. Extensionality and Comprehension. Relations, functions, and well-ordering. Ordinals. Class and transfinite recursion. Cardinals. Well-founded relations, Axiom of foundation, induction, and von Neumann's hierarchy. Relativization, absoluteness, reflection theorems. Gödel's constructible universe L. Axiom of Choice, and Continuum Hypothesis inside L. Po-sets, filters and generic extensions. Forcing. ZFC in generic extensions. Cohen Forcing. Independence of the Continuum Hypothesis. HOD and the Axiom of Choice: independence of the Axiom of Choice.

Keywords

Set Theory, Relative consistency, ZFC, Ordinals, Cardinals, Transfinite recursion, Relativization, Absoluteness, Constructible universe, L, Axiom of Choice, Continuum hypothesis, Forcing, Generic extensions

Learning Prerequisites

Recommended courses

Mathematical logic (or any equivalent course on first order logic). Warning: without a good understanding of first order logic, students tend to get lost sooner orl later.

Important concepts to start the course

- 1st order logic
- basics of proof theory
- Basics of model theory
- Compacity theorem
- Löwenheim-Skolem
- Completeness theorem

Learning Outcomes

By the end of the course, the student must be able to:

- Specify a model of ZFC
- Prove consistency results
- Develop a generic extension
- Argue by transfinite induction

- Decide whether ZFC proves its own consistency
- Formalize the axioms of ZF, AC, CH, DC
- Sketch an inner model
- Justify the axiom of foundation

Teaching methods

Ex cathedra lecture and exercises

Expected student activities

- Attendance at lectures
- Solve the exercises

Assessment methods

- Writen exam (3 hours)
- Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés

Supervision

Office hours   Yes
Assistants   Yes
Forum   Yes

Resources

Bibliography

1. Thomas Jech: Set theory, Springer 2006
2. Kenneth Kunen: Set theory, Springer, 1983
3. Jean-Louis Krivine: Theorie des ensembles, 2007
4. Patrick Dehornoy: Logique et théorie des ensembles; Notes de cours, FIMFA ENS: http://www.math.unicaen.fr/~dehornoy/surveys.html
5. Yiannis Moschovakis: Notes on set theory, Springer 2006
6. Karel Hrbacek and Thomas Jech: Introduction to Set theory, (3d edition), 1999

Ressources en bibliothèque

- Introduction to Set theory / Hrbacek
- Set theory / Jech
- Logique et théorie des ensembles / Dehorny
- Set theory / Kunen
- Notes on set theory / Moschovakis
- Theorie des ensembles / Krivine

Websites

- http://www.hec.unil.ch/logique/

Moodle Link

http://moodle.epfl.ch/course/view.php?id=391

EE-472   **Smart grids technologies**

Paolone Mario

Cursus	Sem.	Type
Energie et durabilité	MA2, MA4	Opt.
Génie électrique et électronique	MA2, MA4	Obl.
Informatique	MA2	Opt.
Mineur en Energie	E	Obl.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	2 weekly

Summary

Learn the technologies and methodologies used in the context smart electrical grids and be able to deploy/implement/test them in a lab environment.

Content

1. Modern monitoring: phasor measurement units technology, synchrophasors extraction processes and time alignement
2. Smart grid communication; reliability, real time and security issues
3. Topology assessment and contingency analysis of power grids
4. Admittance matrix calculus, numerical solution of the load flow problem and state estimation
5. Demand response, real-time and non real-time, forecasting methods applied to renewables and demand
6. Energy management and dispatch plans, the optimal power flow problem

Keywords

Smart grid, power systems

Learning Prerequisites

Required courses

Electric power systems, power distribution networks, TPC/IP Networking

Recommended courses

Signal processing, discrete optimization methods, model predictive control, industrial electronics.

Important concepts to start the course

Understanding of electrical grids and communication networks.

Learning Outcomes

By the end of the course, the student must be able to:

- Design monitoring and control platforms for smart grids
- Test a smart grid
- Implement a smart grid
- Analyze performance of a smart grid

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking
- Manage priorities.
- Use both general and domain specific IT resources and tools

Teaching methods

Ex cathedra, classroom integrated exercises and computer laboratory sessions.

Expected student activities

Attend lectures and labs
Do lab homeworks
Attend test sessions with clickers

Assessment methods

Tests during semester (20%), Written exam (30%) and graded lab reports (50%)

Supervision

Office hours   No
Forum   Yes

Prerequisite for

Master projects in the areas of power systems and energy conversion systems.

EE-593 **Social media**

Gillet Denis

Cursus	Sem.	Type
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	MA2	Opt.
Mineur STAS Chine	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	2
Session	Summer
Semester	Spring
Exam	During the semester
Workload	60h
Weeks	14
Hours	**2 weekly**
Lecture	1 weekly
Project	1 weekly

Summary

The objective is to enable students to critically apprehend the Human Computer Interaction (HCI) challenges associated with the design and the exploitation of social media platforms.

Content

• Social media platforms and the long tail (definition and typology)

• Usability and adoption of social media platforms

• Web 2.0 features and adoption factors

• Privacy, trust and reputation models

• Identities, traces, and Web analytics

• Interplay, between platforms and communities (interdisciplinary perspective)

• Opportunities, requirements and constraints for organization and enterprises

• Participatory design methodologies

• Future ad hoc social applications

Learning Outcomes

By the end of the course, the student must be able to:
• Choose
• Design
• Critique
• Defend

Transversal skills

• Set objectives and design an action plan to reach those objectives.

• Plan and carry out activities in a way which makes optimal use of available time and other resources.

• Use a work methodology appropriate to the task.

• Communicate effectively, being understood, including across different languages and cultures.

• Communicate effectively with professionals from other disciplines.

• Evaluate one's own performance in the team, receive and respond appropriately to feedback.

• Negotiate effectively within the group.

• Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Lectures, invited speakers, individual work and teamwork

Assessment methods

One individual project and one teamwork with combined peer and expert assesment (reports and presentations)

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Bibliography

• Chris Anderson (2006) - The Long Tail: Why the Future of Business is Selling Less of More. New York, NY: Hyperion. ISBN 1-4013-0237-8.

• Joshua Porter - Designing for the Social Web

• Matthew A. Russel - Mining the Social Web: Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites. O¿Reilly 2011

Ressources en bibliothèque

• Designing for the Social Web / Porter

• Mining the Social Web / Russel

• The Long Tail / Anderson

CS-595   **Stage d'ingénieur crédité avec le PDM (master en Informatique)**
Profs divers *

Cursus	Sem.	Type
Informatique	PME, PMH	Obl.
Informatique	MA1, MA2, MA3	Opt.

Langue	
Crédits	0
Session	Hiver, Eté
Semestre	Automne
Examen	Mémoire
Charge	0h
Semaines	
TP	320 hebdo

Remarque

L'étudiant doit effectuer un stage de 8 semaines pendant l'été ou 6 mois après un semestre de Master.

Résumé

Le stage d'ingénieur fait partie intégrante du cursus des étudiants en master. Ils rejoignent des entreprises en Suisse ou à l'étranger pour effectuer un stage ou leur projet de master dans un domaine d'activité où les compétences du futur ingénieur en Informatique sont mises en valeur.

Contenu

Les stages représentent une expérience importante pour les étudiants en leur permettant notamment d'atteindre les buts suivants :
- S'immerger dans le monde professionnel
- Mettre en évidence l'importance du travail en équipe
- Prendre en considération des impératifs d'une entreprise dans ses processus
- Mettre en pratique les connaissances acquises du plan d'études
Les 3 formes de stages suivantes sont possibles dans le cadre du plan d'études master :
- Stage court de 8 semaines pendant l'été uniquement
- Stage long de 6 mois durant le master (l'étudiant est en congé pendant un semestre)
- Projet de master en entreprise (25 semaines)

Mots-clés

Stage; industrie; projet; immersion; connaissances

Compétences requises

Cours prérequis obligatoires

- Avoir effectué un semestre de master pour les étudiants ayant obtenu leur Bachelor à l'EPFL
- Avoir effectué deux semestres de master pour les étudiants venant d'une autre université

Concepts importants à maîtriser

Importance de l'engagement

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Être conscient de l'importance des directives légales et du code éthique de la profession
- Communiquer efficacement et être compris
- Auto-évaluer son niveau de compétence acquise et planifier ses prochaines objectifs
- Gérer ses priorités
- Recevoir et donner du feedback (une critique) et y répondre de manière appropriée

Travail attendu

L'étudiant s'engage à faire son stage avec professionnalisme

Méthode d'évaluation

- Stages courts : évaluation électronique à la fin du stage
- Stages longs : évaluation électronique à la fin du stage
- Projets de Master en entreprise : *voir la fiche "Projet de Master"*

Encadrement

Autres   Superviseur en industrie
                                Superviseur à l'EPFL

Ressources

Polycopiés

Directives des stages : http://ic.epfl.ch/files/content/sites/ic/files/Stages2%20-%20Directives%20FR.pdf

Sites web

- http://ic.epfl.ch/computer-science/internships
- http://stages.epfl.ch/page-93564-en.html

COM-421 — **Statistical neurosciences**

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique	MA2	Opt.
Mineur en Neurosciences computationnelles	E	Opt.
Neurosciences		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Remarque

Pas donné en 2017-18

Summary

In neuroscience, new measurement techniques have permitted to acquire a wealth of experimental data, both scientific and commercial. This class introduces the student to a variety of statistical tools, tailored to the special case of neural data. Students will work with various real data sets.

Content

Examples of the latter include neuromarketing and the control of computer machinery via brain signals. This opens the door for large-scale statistical approaches. The class introduces the student to a variety of statistical tools , tailored to the special case of neural data. An integral part of the class is for the student to work with real data, choosing from a number of data sets and applying the techniques studied in class.

1. Tuning Curves and Receptive Fields (spatio-temporal and spectro-temporal) (5 weeks)
2. Statistical Models, Gaussian Process Factor Analysis (2 weeks)
3. Information-theoretic Techniques (3 weeks)
4. Network Science (2 weeks)

Keywords

Neuroscience, Statistics, Regression, Entropy, Information Theory, Information Measures, Graphical Models

Learning Prerequisites

Required courses

• The class assumes a basic understanding of probability: coin tossing and the standard Gaussian (normal) distribution.

• The class also assumes a basic understanding of linear algebra: vectors, matrices, eigenvalues, eigenvectors.

Learning Outcomes

By the end of the course, the student must be able to:

• Analyze neuroscience data
• Argue in a precise statistical way about neuroscience data
• Interpret neuroscience data
• Justify conclusions about neuroscience data

Teaching methods

Ex cathedra + exercises

Assessment methods

4 homework sets 20%, midterm exam 30% and Matlab project 50%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Here are two books that are related to the class. We do *not* require that you buy these books - but they are recommended reading. (There will be lecture notes for the class.)
1. P. Dayan and L. F. Abbott. *Theoretical Neuroscience*, MIT Press, Cambridge, MA, 2001.In this class, we cover Part I of the book; we will not touch upon Parts II and III.
2. D. Freedman, R. Pisani, and R. Purves. *Statistics*, W. W. Norton & Company, 2007 (4th edition).This is a general-purpose statistics book for all those who do not like excessive mathematical notation,with very good intuitive explanations of many statistical phenomena.

Ressources en bibliothèque

- Statistics / Freedman
- Theoretical Neuroscience / Dayan

Notes/Handbook

Lecture notes will be handed out in class and/or made available on Moodle.

Websites

- http://linx.epfl.ch
- http://linx.epfl.ch/page-70285-en.html

Moodle Link

- http://moodle.epfl.ch

COM-500   Statistical signal and data processing through applications

Ridolfi Andrea

Cursus	Sem.	Type
Informatique	MA2	Opt.
Mineur en Systèmes de communication	E	Opt.
SC master EPFL	MA2, MA4	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly

Summary

Building up on the basic concepts of sampling, filtering and Fourier transforms, we address stochastic modeling, spectral analysis, estimation and prediction, classification, and adaptive filtering, with an application oriented approach and hands-on numerical exercises.

Content

1. Fundamentals of Statistical Signal Processing: Signals and systems from the deterministic and the stochastic point of view. Processing and analysing signals and systems with a mathematical computing language.

2. Models, Methods, and algorithms : Parametric and non-parametric signal models (wide sense stationary, Gaussian, Markovian, auto regressive and white noise signals); Linear prediction and estimation (orthogonality principle and Wiener filter); Maximum likehood estimation and Bayesian a priori.

3. Statistical Signal Processing Tools for Spread Spectrum wireless transmission: Coding and decoding of information using position of pulses (annihilating filter approach); Avoiding interference with GPS (spectral mask and periodogram estimation); Spectrum estimation for classical radio transmissions (estimating frequencies of a harmonic signal).

4. Statistical Signal Processing Tools for the Analysis of Neurobiological Signals: Identification of spikes (correlation-bases methods); Characterization of multiple state neurons (Markovian models and maximum likelihood estimation); Classifying firing rates of neuron (Mixture models and the EM algorithm); Principal Component Analysis.

5. Statistical Signal Processing Tools for Echo cancellation: Adaptive filtering (least mean squares and recursive least squares).

Keywords

Statistical tools, spectral analysis, prediction, estimation, annihilating filter, mixture models, principal component analysis, stochastic processes, adaptive filtering, mathematical computing language (Matlab or similar).

Learning Prerequisites

Required courses

Stochastic Models in Communications (COM-300), Signal Processing for Communications (COM-303).

Recommended courses

Mathematical Foundations of Signal Processing (COM-514).

Important concepts to start the course

Calculus, Algebra, Fourier Transform, Z Transform, Probability, Linear Systems, Filters.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose appropriate statistical tools to solve signal processing problems;
- Analyze real data using a mathematical computing language;
- Interpret spectral content of signals;
- Develop appropriate models for observed signals;
- Assess / Evaluate advantages and limitations of different statistical tools for a given signal processing problem;
- Implement numerical methods for processing signals.

Teaching methods

Ex cathedra with exercises and numerical examples.

Expected student activities

Attendance at lectures, completing exercises, testing presented methods with a mathematical computing language (Matlab or similar).

Assessment methods

- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Background texts

- P. Prandoni, *Signal Processing for Communications*, EPFL Press;
- A.V. Oppenheim, R.W. Schafer, *Discrete Time Signal Processing*, Prentice Hall, 1989;
- B. Porat, *A Course in Digital Signal Processing*, John Wiley & Sons,1997;
- C.T. Chen, *Digital Signal Processing*, Oxford University Press;
- D. P. Bertsekas, J. N. Tsitsiklis, *Introduction to Probability,* Athena Scientific, 2002 (excellent book on probability).

More advanced texts

- L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*, Springer-Verlag, 1988;
- A.N. Shiryaev, *Probability*, Springer-Verlag, New York, 2nd edition, 1996;
- S.M. Ross, *Introduction to Probability Models*, Third edition, 1985;
- P. Bremaud, *An Introduction to Probabilistic Modeling*, Springer-Verlag, 1988;
- S.M. Ross, *Stochastic Processes*, John Wiley, 1983;
- B. Porat, *Digital Processing of Random Signals*, Prentice Hall,1994;
- P.M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, 1993;
- P. Stoïca and R. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, 1997.

Ressources en bibliothèque

- Probability / Shiryaev
- Stochastics Processes / Ross

- Discrete Time Signal Processing / Oppenheim
- Introduction to Spectral Analysis / Stoïca
- Digital Processing of Random Signals / Porat
- Introduction to Probability / Bertsekas
- Introduction to Probability Models / Ross
- Signal Processins for Communications / Prandoni
- An Introduction to Probabilistic Modeling / Bremaud
- A Course in Digital Signal Processing / Porat
- Optimal and Adaptive Signal Processing / Clarkson
- Digital Signal Processing / Chen
- Introduction to Hilbert Spaces with Applications / Debnath

Notes/Handbook

- Slides handouts;
- Lecture notes;
- Collection of exercises.

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

CS-448   **Sublinear algorithms for big data analysis**
Kapralov Mikhail

Cursus	Sem.	Type
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly

Summary

In this course we will define rigorous mathematical models for computing on large datasets, cover main algorithmic techniques that have been developed for sublinear (e.g. faster than linear time) data processing. We will also discuss limitations inherent to computing with constrained resources.

Content

The tentative list of topics is:

Streaming: given a large dataset as a stream, how can we approximate its basic properties using a very small memory footprint? Examples that we will cover include statistical problems such as estimating the number of distinct elements in a stream of data items, finding heavy hitters, frequency moments, as well as graphs problems such as approximating shortest path distances, maximum matchings etc.;
Sketching: what can we learn about the input from a few carefully designed measurements (i.e. a `sketch') of the input, or just a few samples of the input? We will cover several results in sparse recovery and property testing that answer this question for a range of fundamental problems;
Sublinear runtime: which problems admit solutions that run faster than it takes to read the entire input? We will cover sublinear time algorithms for graph processing problems, nearest neighbor search and sparse recovery (including Sparse FFT);
Communication: how can we design algorithms for modern distributed computation models (e.g. MapReduce) that have low communication requirements? We will discuss graph sketching, a recently developed approach for designing low communication algorithms for processing dynamically changing graphs, as well as other techniques.

Keywords

streaming, sketching, sparse recovery, sublinear algorithms

Learning Prerequisites

Required courses

Bachelor courses on algorithms, complexity theory, and discrete mathematics

Important concepts to start the course

Discrete probability; mathematical maturity

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class

- Analyze space/time/communication complexity of randomized algorithms

- Prove space/time/communication lower bounds for variations of problems discussed in class

• Choose an appropriate algorithmic tool for big data problem at hand

Teaching methods

Ex cathedra, homeworks, final

Assessment methods

Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

CS-550 **Synthesis, analysis and verification**

Cursus	Sem.	Type
Computer engineering minor	E	Opt.
Informatique et communications		Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly

Remarque

pas donné en 2017-18

Summary

The course presents theory, algorithms, and tools for reasoning about computer systems, including techniques for software and hardware verification and synthesis.

Content

Motivation:
Tools for automated analysis and verification of software can improve reliability of software that we use every day. The underlying techniques are also used for compiler optimizations and program understanding. In recent years, new algorithms and combinations of existing techniques have made such tools more effective than in the past. This course will give an overview of basic techniques, as well as the recent advances that made this progress possible.
In many years the course also contains guest lectures presenting recent research results.

Topics covered include:

- Logic and relational program semantics
- Verification condition generation and Hoare logic
- Synthesis of programs from relations
- Abstract interpretation and data flow analysis
- Predicate abstraction
- Modular verification
- Decision procedures, SMT solvers, and resolution-based provers

Learning Prerequisites

Required courses

Theoretical computer science and discrete mathematics course, or equivalent background and fluency in discrete mathematics and introductory theoretical computer science concepts (e.g. M. Sipser textbook) Functional programming in Scala, or ability to pick up Scala quickly (students knowing Haskell or ML generally have no trouble).

Recommended courses

The knowledge of mathematical logic and combinatorial optimization is beneficial

Learning Outcomes

By the end of the course, the student must be able to:
- Formalize program correctness
- Prove correctness of programs on paper
- Sketch an automated verification algorithm
- Interpret results of verification systems
- Create a simple program verifier
- Construct a constraint solver
- Systematize approaches to software correctness
- Choose an appropriate technique for improving software reliability

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- Respect the rules of the institution in which you are working.
- Demonstrate a capacity for creativity.
- Make an oral presentation.
- Summarize an article or a technical report.
- Write a scientific or technical report.
- Communicate effectively with professionals from other disciplines.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.

Teaching methods

Ex catedra
Exercise sessions
Practical work on projects under supervision of teaching assistants

Expected student activities

Attending lectures
Exercises in class
Homeworks
Mid-term exam
Practical project on modifying a verification system

Assessment methods

- 30% common project in first part of semester (in stages and feedback after each, but grade only after all of them)
- 40% quiz in 2nd part of semester
- 30% individual projects by the project deadline

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

The Calculus of Computation: Decision Procedures with Applications to Verification. **Bradley**, Aaron R., **Manna**, Zohar, Springer, 2007. ISBN 978-3-540-74113-8.

Ressources en bibliothèque

• The Calculus of Computation / Bradley

Notes/Handbook

http://lara.epfl.ch/w/sav13:top

Websites

• http://lara.epfl.ch/w/sav

Videos

• https://www.youtube.com/watch?v=rm_kqt61JQ8

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

COM-407   **TCP/IP networking**

Le Boudec Jean-Yves

Cursus	Sem.	Type
Génie électrique et électronique	MA1, MA3	Opt.
Information security minor	H	Opt.
Informatique	MA1, MA3	Opt.
Mineur en Systèmes de communication	H	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly

Summary

In the lectures you will learn and understand the main ideas that underlie and the way networks are built and run. You will be able to apply the concepts to the smart grid. In the labs you will exercise practical configurations.

Content

LECTURES: 1. The TCP/IP architecture 2. Layer 2 networking; Bridging; the Spanning Tree Protocol. Bellman Ford. 3. The Internet protocol versions 4 and 6 4. The transport layer, TCP, UDP, sockets 5. Distance vector, link state routing. Optimality of routing. Interdomain routing, BGP. 6. Congestion control principles. Application to the Internet. The fairness of TCP. Flow based networking. Reservations for quality of service. 7. Hybrid constructions and tunnels, MPLS, VPNs. VPNs. 8. Selected advanced topic.
LABS: 1. Configuration of a network, virtual machines and GNS3 2. MAC; NATs and troubleshooting 3. Socket programming 4. Interior routing 5. Congestion control and flow management 6. BGP

Keywords

TCP/IP
Computer Networks

Learning Prerequisites

Required courses

A first programming course

Learning Outcomes

By the end of the course, the student must be able to:
- Run and configure networks
- Understand the main ideas that underlie the Internet
- Write simple communicating programs
- Use communication primitives for internet applications or in the smart grid

Transversal skills

- Access and evaluate appropriate sources of information.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Lectures with clickers

Labs on student's computer and in the Internet Engineering Workshop

Expected student activities

Participate in lectures
Participate in graded clicker test every other week
Make one lab assignment every other week, including handing in a written report
Optional: research exercise: gather information about a specific topic and explain it to class

Assessment methods

Theory grade = max(40% clicker test + 60% final exam, final exam)
Practice grade = average of labs
Final grade = harmonic mean of theory grade and practice grade.
The research exercice may give a bonus of at most 0.5 points in 1-6 scale.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

"Computer Networking : Principles, Protocols and Practice", O. Bonaventure, open source textbook,
http://inl.info.ucl.ac.be/CNP3

Ressources en bibliothèque

• Computer Networking / Bonaventure

Notes/Handbook

Slides are on moodle

Websites

• http://moodle.epfl.ch/course/view.php?id=523

Moodle Link

• http://moodle.epfl.ch/course/view.php?id=523

Videos

• http://moodle.epfl.ch/course/view.php?id=523

CS-410	**Technology ventures in IC**

Cursus	Sem.	Type
Data Science	MA2	Opt.
Informatique	MA2	Opt.
Mineur STAS Chine	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly

Remarque

pas donné en 2017-18

Summary

This hands-on class gives graduate students in IC interested in startups the opportunity to learn and put in practice the fundamental skills required to assess a technology concept in the context of a business opportunity. This class is focused only on business opportunities where high-technology

Content

Working in teams, students will learn the fundamentals of:

- *Opportunity assessement*
- *Customer development and validation*
- *Business model alternatives*
- *Intellectual Property*
- *Strategy and Financial planning*
- *Go-to-market, launch, and growth*

This is a hands-on class where students start the class with their own technology venture concept (e.g. the work done as part of their PhD, or some well-formed idea, maybe with a prototype).   During the class, they convert their concept into a integrated business plan.

Keywords

Entrepreneurship, startups, technology transfer, intellectual property

Learning Prerequisites

Required courses

- *None – but available to MS and Ph.D. students only*

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze a business plan
- Create a business plan

Teaching methods

- Short ex-cathedra presentations of each topic
- Hands-on seminar with many short student presentations
- Presentations from invited guests, in particluar industry executives and entrepeneurs
- Discussion and case studies

Assessment methods

- In-class participation (30%)
- In-class presentations (30%)
- Final pitch (40%)

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

CS-455   Topics in theoretical computer science

Cursus	Sem.	Type
Data Science	MA1	Opt.
Informatique	MA1, MA3	Opt.
Mineur en Informatique	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly

Remarque

pas donné en 2017-18

Summary

The students gain an in-depth knowledge of several current and emerging areas of theoretical computer science. The course familiarizes them with advanced techniques, and develop an understanding of fundamental questions that underlie some of the key problems of modern computer science.

Content

- Examples of topics to be covered include:

- Streaming: given a large dataset as a stream, how can we approximate its basic properties using a very small memory footprint? Examples that we will cover include statistical problems such as estimating the number of distinct elements in a stream of data items, finding heavy hitters, frequency moments, as well as graphs problems;
- Sketching and sampling: what can we learn about the input from a few carefully designed measurements (i.e. a `sketch') of the input, or just a few samples of the input? We will cover results in sparse recovery and property testing that answer this question for several fundamental problems;
- Sublinear runtime: which problems admit solutions that run faster than it takes to read the entire input? Examples include sublinear time algorithms for graph processing problems, nearest neighbor search and Sparse FFT;
- Communication: how can we design algorithms for modern distributed computation models (e.g. MapReduce) that have low communication requirements? We will discuss graph sketching, a recently developed approach for designing low communication algorithms for processing dynamically changing graphs.

Keywords

streaming, sketching, sparse recovery, sublinear algorithms

Learning Prerequisites

Required courses

Bachelor courses on algorithms, complexity theory, and discrete mathematics.

Learning Outcomes

By the end of the course, the student must be able to:
- Design efficient algorithms for variations of problems discussed in class;
- Analyze formally space/time/communication complexity of randomized algorithms

- Prove space/time/communication lower bounds for variations of problems discussed in class;
- Select appropriately algorithmic tool for big data analysis problem at hand

Teaching methods

Ex cathedra, homeworks, reading

Expected student activities

Attendance at lectures, completing exercises, reading written material

Assessment methods

- Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Others	Electronique forum : Yes

Resources

Bibliography

There is no textbook for the course. Notes will be posted on the course website.

Ressources en bibliothèque

- Randomized Algorithms / Motwani

CS-444	**Virtual reality**
	Boulic Ronan

Cursus	Sem.	Type
Data Science	MA2	Opt.
Humanités digitales	MA2	Opt.
Informatique	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly

Summary

The goal of VR is to embed the users in a potentially complex virtual environment while ensuring that they are able to react as if this environment were real. The course provides a human perception-action background and describes the key techniques for achieving efficient VR applications.

Content

The first lectures focus more on the technical means (hw & sw) for acheiving the hands-on sessions:

- Visual display (CAVE and stereoscopy)
- Interaction devices and sensors
- Software environment

The proportion of more theoretical VR and Neuroscience background increases over the semester:

- Key Human perception abilities, Cybersickness, Immersion, presence and flow
- Basic 3D interaction techniques: Magic vs Naturalism
- The perception of action
- Haptic interaction
- What makes a virtual human looking alive ?
- Motion capture for full-body interaction
- VR, cognitive science and true experimental design

Keywords

3D interaction, display, sensors, immersion, presence

Learning Prerequisites

Required courses

(CS 341) Introduction to Computer Graphics

Recommended courses

(CS 211) Introduction to Visual Computing

Important concepts to start the course

from Computer Graphics:
- perspective transformations
- representation of orientation
- 3D modelling hierarchy
- matrix algebra: translation, orientation, composition

Learning Outcomes

By the end of the course, the student must be able to:

- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersions, presence and flow
- Give an example of applications of VR in different industrial sectors
- Choose a method of immersion suited for a given 3D interaction context
- Explain the possible causes of cybersickness in a given VR system configuration
- Design a VR system involving 3D interactions

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Ex cathedra + Hands-on sessions on VR devices in the first half of the semester,
a mini-project in groups on personal laptops will have to integrate various components of 3D real-time interaction. the group will negociate their project proposal with the course responsible TA who will assess whether it meets the key specifications and is original enough. The proposal can include the use of some VR devices that the IIG research group will lend on a first-come/first-served basis.

Expected student activities

exploit citation analysis tools to evaluate a scientific paper
combine libraries to produce an original 3D interaction
experiment the hands-on practical work in the lab
synthesize the knowledge acquired in course and hands-on in the quizzes and final oral

Assessment methods

Throughout semester: 4 Hand-on sessions (4%), 2 Quizzes (10%), 1 paper citation study (16%), 1 mini-project (40%), 1 oral (30%)

Supervision

Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Course notes will be updated and made available after each course, with links to key sites and on-line documents
- J. Jerald, The VR Book, ACM Press 2015
- Parisi, Learning Virtual Reality, O'Reilly 2015
- Le Traité de Réalité Virtuelle (5 vol.) Presses des Mines, ParisTech, 2006-2009, available on-line, free for student upon registration.
- Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Ressources en bibliothèque

- 3D User Interfaces: Theory and Practice / Bowman
- Le Traité de Réalité Virtuelle / Fuchs
- The VR Book / Jerald
- Learning Virtual Reality / Parisi

Notes/Handbook

pdf of slides are made visible after the ex-cathedra courses

Websites

- http://www.thevrbook.net/

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=6841

INDEX PAR COURS

INDEX PAR COURS

INDEX PAR COURS

INDEX PAR ENSEIGNANTS

INDEX PAR ENSEIGNANTS

INDEX PAR ENSEIGNANTS

INDEX PAR ENSEIGNANTS