The Impact of Students' Pre-Knowledge on Learning Computer Programming

MARKO HÖLBL and LILI NEMEC ZLATOLAS, University of Maribor

Students taking courses of computer programming have different pre-knowledge on the topic, which not only includes their knowledge of a specific computer programming language but also their ability for algorithmic thinking and the understanding of the concept of computer programming. At the beginning of the course, they were asked to self-evaluate their knowledge of computer programming and take a pre-course test of their knowledge in JavaScript. In the scope of the research, we have compared the results of the pre-course test of students with technical and general High School education, as well as the results they achieved at the post-course test. Altogether, 55 students collaborated in both pre and post-course tests. The results indicate that students with technical and general High School backgrounds scored similarly on a pre-course test of computer programming and that students with general High School backgrounds acquired more knowledge in the course compared to the ones with a technical High School background. The results also indicate that the students with lower self-evaluation of computer programming have acquired more knowledge during the course compared to the ones with higher self-evaluation scores.

1. INTRODUCTION

Learning the principles of computer programming is an essential skill for an IT professional. Teaching computer programming is a challenge, as the algorithmic way of thinking is uncommon for people [Matthew et al. 2007, Futschek 2006, Knuth 1985, Sleeman 2002]. There are significant differences between experts and novices in computer programming, as well as many challenges on how to teach computer programming in order for the students to learn as fast as possible [Cooper et al. 2000, Fowler and Cusack 2011]. Becoming a good programmer is a cumbersome process, requiring a lot of studying and practising.

Students with different secondary education level backgrounds are joining Universities to study Informatics and Computer Science or an ICT related study. A programming course usually serves as a fundamental course in any ICT-related Study Programme, due to its importance in understanding the concept of ICT systems. Additionally, the development of ICT solutions and/or services requires knowledge of computer programming.

Many students come to the university with little or no pre-knowledge of programming and often tend to overestimate their knowledge.

The studies have shown that pre-knowledge can have an effect on students so that they believe they already have enough knowledge on the topic and do not need to learn any more [McMillan and Hearn 2008, Sewell 2002]. This might lead to the effect that such students do not achieve the level required by university standards. Additionally, previous studies also employed different approaches to teaching programming [Janzen and Saiedian 2008, Kalelioğlu 2015].

The authors acknowledge the financial support from the Slovenian Research Agency (Research Core funding No. P2-0057) and from the European Union's Horizon 2020 Research and Innovation program under the CyberSec4Europe project (GA No. 830929). We would also like to thank the participants of this research project.
Author's address: M. Hölbl and L. Nemec Zlatolas, Faculty of Electrical Engineering and Computer Science, Koroska cesta 46, 200 Maribor, Slovenia; email: marko.holbl@um.si, lili.nemeczlatolas@um.si

In: Z. Budimac and B. Koteska (eds.): Proceedings of the SQAMIA 2019: 8th Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, Ohrid, North Macedonia, 22–25. September 2019. Also published online by CEUR Workshop Proceedings (http://ceur-ws.org, ISSN 1613-0073)

In this work, we present a study where we analysed if students with different secondary education level backgrounds, thus with different pre-knowledge of computer programming and their self-perception of this knowledge, influenced their results in learning computer programming. The analysis was performed with active students before taking the course, and after they had taken the course lessons and completed the practical laboratory work. The experiment was conducted with students who took the course Programming for Media, which involves the basics of JavaScript programming.

We set up the following research question: How does the pre-knowledge of computer programming affect the knowledge of computer programming after taking lectures from the computer programming course?

The structure of the paper is as follows. The description of research methods is provided in Section 2 and the main contribution of the paper in Section 3, where the results and a discussion are given. Finally, the conclusions are presented in Section 4.

2. RESEARCH ON STUDENTS' KNOWLEDGE OF COMPUTER PROGRAMMING IN JAVASCRIPT

We have used a test type of evaluation of pre and post knowledge of students in programming. Further details are presented in the following sub-sections.

2.1 Data collection and participants

At the beginning of the semester, students attending the Programming for Media Course were asked to fill in a test to assess their skills on computer programming in JavaScript. After the course was finished and they had attended lectures as well as the exercises of the course, their knowledge and acquired skills in computer programming in JavaScript were tested again. The sample of demographics is presented in Table I. Altogether, 55 students collaborated in both parts of the survey – the pre-knowledge test and the test after they obtained knowledge at lectures and exercises. Most of the students were in their first or second year of studies. There were more female students collaborating in the survey, and the majority of students were enrolled in the Study Programme Media Communications. The previous secondary education (High School) of students was either general or technical.

Table I. Sample of demographics (n=55).

VARIABLE	SAMPLE RESULTS
Gender	Male 43.6% Female 56.4%
Study Programme	Media Communications 74.5% Informatics and Technologies of Communication 25.5%
Previous education	General 60.0% Technical 40.0%

2.2 Measures

Measurement items were tested with a 7-point Likert-scale ranging from 1 to 7. The measurement items are presented in Table II. There were altogether 25 questions in the survey. To connect the pre-course test of each student and the post lectures test, we used a unique ID number that a student had to enter when filling in the pre-course test on his/her knowledge and on the exam at the

end of the course. The first part of the test (demographical questions and questions 1-11 in Table II) were only asked in the pre-course test. In the post-course test (after the students had already done the tasks and taken the lectures) only the questions 12-18 were asked. The questions were chosen to suit the lectures concept. To connect the pre-knowledge of each student and the grade for Database Modelling, we used a unique ID number that a student had to enter when filling in the self-evaluation questionnaire on his/her knowledge and on the exam at the end of the course.

Table II. Measurement of variables.

QUESTIONS	**POSSIBLE ANSWERS**	**Present in pre-course test**	**Present in post-course test**
Have you already attended courses on computer programming before?	Y/N	Y	N
I have a lot of pre-knowledge on computer programming. I know how to use the JavaScript computer programming language. I know how to use some computer programming language. I know how to use variables. I know how to use arrays. I know how to use conditional expressions / statements. I know how to use loops for computer programming. I know how to use functions in computer programming. I know how to use Objects in computer programming. I know how to use DOM.	1 – Strongly disagree 2 – Disagree 3 – More or less disagree 4 – Undecided 5 – More or less agree 6 – Agree 7 – Strongly agree	Y	N
What is a variable?	A box into which we save the data. A box into which we save unchangeable data. JavaScript expression. JavaScript number. I don't know.	Y	Y

If the user enters "10" in a prompt window, what will be the result?
`var x = Number (prompt("Insert value:"));`
`var res = x * 3 - 1;`
`res += 5;`
`console.log(res);`
	24
	34
	4
	Undefined
		Y	Y

If the user enters "10" in a prompt window, what will be the result?
`var x = Number (prompt("Insert value:"));`
`if (x <= 10) {`
`if (x >= 5) {`
`console.log ("A");}`
`console.log("B");} else {`
`console.log("C");}`
	"A"
	"B"
	"C"
	"A" in "B"
		Y	Y

Which operator needs to be in the place where "???" is now so that the expression makes sense? `var num1 = Number (prompt("Insert number 1:"));`	AND operator: && OR operator: || NOT operator: ! No operator.	Y	Y

QUESTIONS	POSSIBLE ANSWERS	Present in pre-course test	Present in post-course test

`var num2 = Number (prompt("Insert number 2:"));`
`if((num1 < 0) ??? (num2 < 0)){`
`console.log("At least one of the numbers is negative.");}`
`else{console.log("Both numbers are positive or 0."); }`			

How many times will the text appear in the program?
 	
`var i = 0;`
`while (i <= 4) {`
`console.log ("Hello!");`
`i++; }`
	Once
	4 times
	5 times
	An infinite number of times
		Y	Y

How many times will the text appear in the program?
 	
`for(var i = 1; i < 5; i ++) {`
`console.log ("What is up?");}`
	Once
	4 times
	5 times
	An infinite number of times
		Y	Y

If we do not know how many times the loop should repeat, which loop should we use?	While loop For loop Whichever – For or While loop.	Y	Y

The correct answer is marked with bold in the questions 12-18.

3. DATA ANALYSIS AND RESULTS

After collecting the data using the test, we have used SPSS for data analysis in this study. The results are presented in the following sub-sections.

3.1 Testing the programming pre-knowledge of students with different educational backgrounds

First, we compared the students with technical and general High School background. We conducted the pre-course test of knowledge with questions 12-18, which are given in Table II. On average, both the students with technical and general high school educational background scored similarly on a 7 question test for pre-knowledge of computer programming in JavaScript. The average for a group with a technical High School background was 3.68; while students with a general High School background scored a bit better in the pre-course test that is 3.79 points on average out of 7. A more detailed analysis is presented in Table III.

Table III. Comparison of pre-knowledge between students with different High School backgrounds.

High school education	Pre-course test score (0-7)								TOTAL	AVERAGE
	.00	1.00	2.00	3.00	4.00	5.00	6.00	7.00		
Technical High School	0	5	2	3	5	2	2	3	22	3.68
General High School	2	4	5	3	3	8	6	2	33	3.79
TOTAL	2	9	7	6	8	10	8	5	55	3.75

Additionally, we analysed how much knowledge students with general and technical High School backgrounds acquired during the course by calculating the difference in the pre and post-course tests. These were the same in both cases (questions 12-18 in Table II). Altogether, 7 points could be achieved in repeated tests. If a student scored more points in the post-course test than in the pre-course test, the value was positive. For example, if a student scored 4 points in the pre-course test and 6 points in the post-course test, the difference would be +2 points. As Figure 1 indicated, students with a technical High School, background improved on average by 1 point, and students with a general High School background on average by 2 points between the pre- and post-course tests.

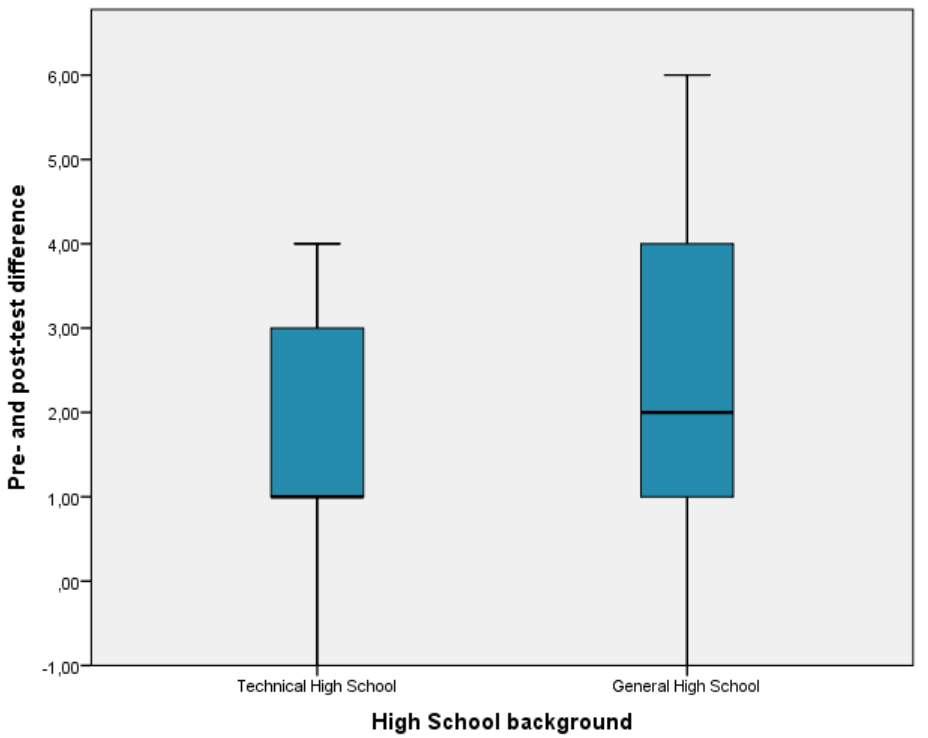

Fig. 1. A box plot of the difference in pre- and post-course tests between groups with different High School backgrounds.

3.2 Comparison of pre- and post-course test results from students with different pre-knowledge on programming

An analysis of the variance between different groups was conducted next. Questions 2 to 11 indicate the level of knowledge of JavaScript programming according to the self-evaluation of students. The following 7 questions test basic users' knowledge on JavaScript programming (questions 12 – 18 given in Table II). After calculating the difference, we ran a Univariate analysis of Variance and compared the different groups.

Next, we divided students based on their self-evaluation in computer programming. Group 1 had an average lower than 4 out of 7 when doing the self-evaluation (questions 2-11), while group 2 self-assessed their computer programming knowledge higher than 4 out of 7. As can be seen from Table IV, students who had lower pre-knowledge of computer programming improved their score on

average by 2.53 points after taking lectures, compared to the ones with higher pre-knowledge. Those improved their score just by 0.91 points. However, students with higher pre-knowledge scored better in both cases – in the pre-course test and in the post-course test. Yet, in the post-course test, the difference was not as high. The results of the univariate analysis of variance also indicate that the significance of Levene's test [David et al. 2006] is <.005, which rejects the null hypothesis, where we tested if the error variance of the dependent variable was equal across the groups. This indicates a significant difference between groups with low pre-knowledge and high pre-knowledge in computer programming. We conducted another test, the "Test of Between-Subjects Effects with dependent variable Difference in pre- and post-course tests", and the difference between the two groups was also significant, and lower than 0.005. Furthermore, the partial eta squared is .210, which means that 21% of the variance is explained in the dependent variable.

Table IV. Univariate Analysis of Variance of groups with low and high pre-knowledge.

Descriptive statistics for groups					
	Pre-course test		**Post-course test**		**N**
	Mean	**SD**	**Mean**	**SD**	32
Group 1: low pre-knowledge	2.75	1.97	5.28	1.20	32
Group 2: high pre-knowledge	5.13	1.25	6.04	1.02	23
Descriptive statistics with dependent variable Difference in pre- and post-course tests					
	Mean (difference)		**Standard Deviation**		**N**
Group 1: low pre-knowledge	2.53		1.81		32
Group 2: high pre-knowledge	0.91		1.16		23
Levene's Test of Equality of Error Variances with dependent variable Difference in pre- and post-course tests					
F	**Df1**		**Df2**		**Sig.**
7.473	1		53		.008
Test of Between-Subjects Effects with dependent variable Difference in pre- and post-course tests					
Mean Square	**F**		**Sig.**		**Partial Eta Squared (R squared)**
35.042	14.092		.000		.210

Fig. 2 depicts the box plot of the two different groups and the differences between the pre and the post-course tests. As can be concluded from the boxplot, some of the students have even scored less in the post-course test than in the pre-course test. Some with low pre-knowledge even scored up to 6 points (out of 7) more in the post-course test, while the others with high pre-knowledge scored mostly up to 3 points more in the post-course test in comparison to the pre-course test.

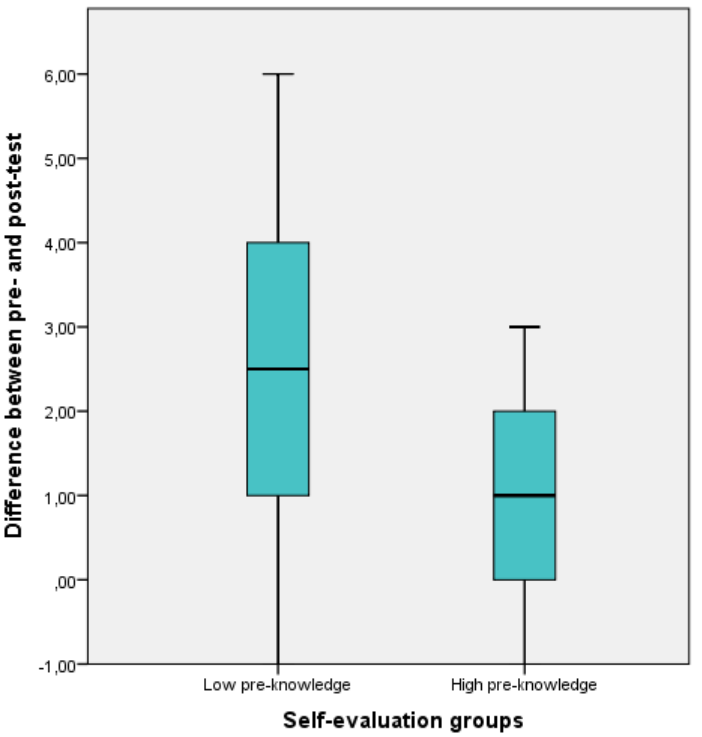

Fig. 2. A box plot of the difference between groups for the pre-and post-course test.

3.3 Discussion

In this study, we investigated how the pre-knowledge of computer programming affects the knowledge of computer programming after taking lectures from the computer programming course. We found that the students with low pre-knowledge in computer programming gained more knowledge during the lessons than the ones with high pre-knowledge of programing. Additionally, students with a more technical background in computer programming did not present better knowledge of computer programming in the pre-course test than students with no technical backgrounds. Students having luck in the pre-course test, since the answers were a, b, c, d and they could have chosen the right one, which could explain this. However, it would be expected that, if they did not know the answer, they would choose the "don't know" option. Students from general High School backgrounds also learned more while taking the lessons then the students from technical High Schools.

4. CONCLUSION

At the Faculty of Electrical Engineering and Computer Science at the University of Maribor, students were taking the course Programming for Media in the first or second year of their studies. We asked the students at the beginning of the course to self-evaluate their knowledge of Programming in JavaScript, and, at the same time, test their computer programming skills using a questionnaire. After taking the course, the students again took a test of their computer programming skills. Our results indicate that the students with no, or very little, pre-knowledge of computer programming had acquired much knowledge from the basic course of computer programming, and

had achieved almost the same results as students with a lot more pre-knowledge. We could conclude that the self-esteem of the students with pre-programming skills is not a positive thing, because the students put less effort into the course. Our statistical analysis showed that students with technical and general High School backgrounds scored similarly on a pre-course test of computer programming, which was quite surprising. We expected that students with a technical background would have more knowledge of computer programming. Students with general High School backgrounds also acquired more knowledge while taking the course than students with a technical High School background. The limitation of this study is a small sample, but the study could be tested on a larger sample.

Additionally, it would be expected that students with a technical background would acquire more knowledge after attending the lessons of the Programming for Media. However, our results indicate the opposite.

REFERENCES

Matthew Butler, Matthew Butler, and Michael Morgan. 2007. Learning challenges faced by novice programming students studying high level. In *Proceedings of the 24th ascilite Conference, 99–107.*

Stephen Cooper, Wanda Dann, and Randy Pausch. 2000. Developing Algorithmic Thinking With Alice. In *Proceedings of the* Isecon, 506–539.

I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, and H. B. Mann. 2006. Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling. DOI:https://doi.org/10.2307/2984135

Adrian Fowler and Brian Cusack. 2011. Kodu Game Lab: Improving the motivation for learning programming concepts. In Foundations of Digital Games, 238–240. DOI:https://doi.org/10.1145/2159365.2159398

Gerald Futschek. 2006. Algorithmic Thinking: The Key for Understanding Computer Science. In International conference on informatics in secondary schools-evolution and perspectives, 159–168. DOI:https://doi.org/10.1007/11915355_15

David Janzen and Hossein Saiedian. 2008. Test-driven learning in early programming courses. In ACM SIGCSE Bulletin, 532. DOI:https://doi.org/10.1145/1352322.1352315

Filiz Kalelioğlu. 2015. A new way of teaching programming skills to K-12 students: Code.org. Comput. Human Behav. 52, (November 2015), 200–210. DOI:https://doi.org/10.1016/J.CHB.2015.05.047

Donald E. Knuth. 1985. Algorithmic Thinking and Mathematical Thinking. Am. Math. Mon. 92, 3 (1985), 170. DOI:https://doi.org/10.2307/2322871

J H McMillan and Jessica Hearn. 2008. Student self-assessment: The key to stronger student motivation and higher achievement. Educational Horizons 87: 40–49

Derek Sleeman. 2002. The challenges of teaching computer programming. Commun. ACM 29, 9 (2002), 840–841. DOI:https://doi.org/10.1145/6592.214913