How to Improve Programming for Novices

Jasna Hamzabegović, Ph.D.
University of Bihać
Pedagogical Faculty/Department of Mathematics and Informatics Bihać
Bosnia and Herzegovina

Damir Kalpić, Ph.D.
University of Zagreb
Faculty of Electrical Engineering and Computing Zagreb
Croatia

Emina Junuz, Ph.D.
University "Džemal Bijedić" in Mostar
Faculty of information technologies
Bosnia and Herzegovina

Abstract

Due to high job demand and high offered salaries, programming is one of the most attractive careers today. Ever more young people enrol in information technology (IT) and computer science (CS) courses. However, programming is a complex and difficult activity and the attrition from related courses continues to be significant. Introductory programming courses traditionally have high failure rates. As this subject tends to be the core of IT and CS curricula, it can become a roadblock for many students preventing them to continue their university studies. Is it really so difficult to learn programming and what do students, as future professional programmers, think about that? We conducted a survey among novices – future professional programmers after the semester in which students had attended an introductory course in programming. Everything indicates that programming is a challenging and difficult activity. In this paper, we show that the students who have regularly attended lectures, tutorials and workshops still can pass the examination in the introductory programming course, indicating that teaching activities cannot be substituted by individual e-learning alone.

Keywords — introductory programming course, learning difficulty, attrition from CS courses, survey, statistical test

Introduction

Programming is an undisputed job of the future. Software surrounds us in every minute of our lives, and people who create it are not going to be out of job in foreseeable future.

Nevertheless, this is a complex and difficult intellectual activity. Programming contains elements of art, science, mathematics and construction. It requires abstract thinking and formal mode of expression. Programming is a creative process.
It is not easy to learn the programming. Students at IT study struggle to learn it. Research shows that for many students the problems appear already in the initial learning phase, when trying to understand and apply the abstract concepts of programming, such as complex control structures and loops for creating algorithms to solve specific problems [1].

In a survey of failure rates for introductory programming courses from 2007 [2], it was found that the average failure rate in the introductory programming course had been 33%. For universities outside the U.S., the result was 41%. Quite a few major European universities reported failure rates exceeding 50%.
As this course tends to be the core of IT and Computer Science courses, it can be a roadblock for many students to continue their university studies.
Is it difficult to learning programming? What do students, as future professional programmers, think about this issue?

Research

We conducted a survey among students – future professional programmers [3].

The participants

The participants in our study were:
• students of the Faculty of Electrical Engineering – Computer Science Department
• students of the Faculty of Information Technology and
• students of the Pedagogical Faculty at the Department of Mathematics and Computer Science.
The survey was done in February 2013. This is one semester after they got an introductory course in programming.

Previous knowledge and involvement

From all the students who participated in the survey (Fig. 1), 49% of them had no previous knowledge of programming language they should learn during their studies, while 22% had little, 25% moderate and 4% extensive knowledge.

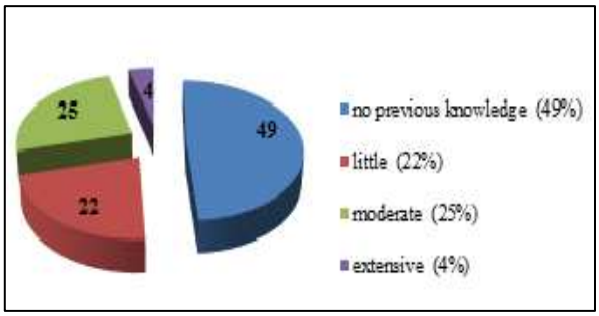

Fig. 1. Knowledge of programming language

Although 93% of them attended lectures, tutorials and workshops regularly, and spent as much or twice as much time (57%) practicing at home as they did in tutorials/workshops, or slightly less than at tutorials/workshops (42%), and only 1% never practiced independently, midterm test results do not follow the degree of students' involvement (Fig. 2). In fact, only

67% of students passed all the midterms. This indicates that the programming is difficult for beginners.

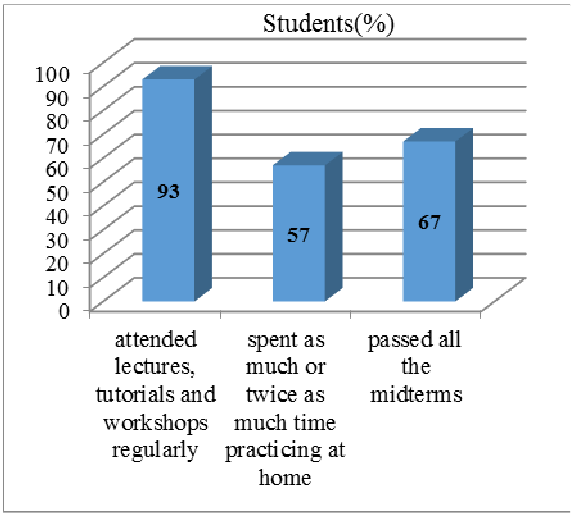

Fig. 2. Students' involvement and midterm test achievement

Figure 3 shows the survey results, which indicate that the programming is difficult and rather abstract, and that the abstraction was difficult to understand.
To acquire the abstraction inherent to programming, for 22% of students it took half semester, 19% more than half semester, 42% the whole semester and 7% of them have never understood it. Only 10% understood it immediately.

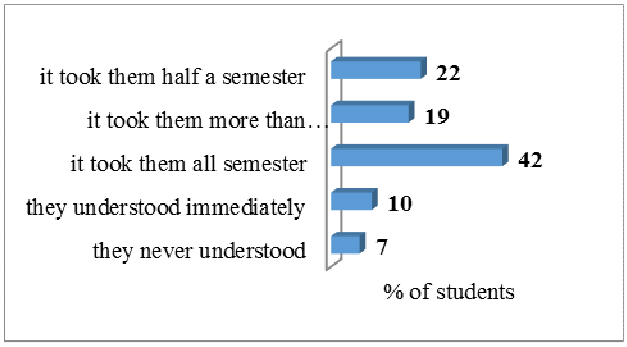

Fig. 3. Period of time to master the programming - to acquire the abstraction inherent to programming

What students think

Of the surveyed students, 53% of them prefer not to learn syntax. In addition, most students–respondents thought that programming would be easier in the development environment. Therefore, 76% of them would prefer to use development environment (e.g. Visual Studio). From Fig. 4, we can see that 93% of respondents stated the fact that programming required great mental activity, 98% indicated that it required abstract thinking, 95% indicated a good knowledge of programming language syntax was required and 88% stated formal ways of expressing ware required.

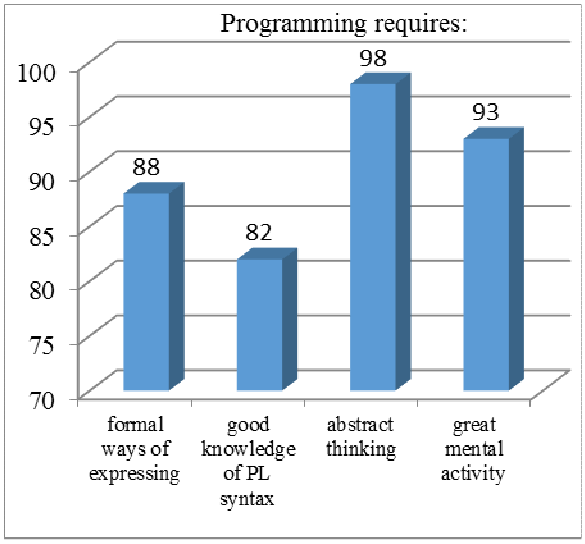

Fig. 4. What students(%) think

Mastered syntax

Of the entire knowledge that students have gained, 33% of had been acquired by writing programs, 15% by reading manuals and 19% through required readings. Even in the late 80s, researches conducted among students, which showed that students (in different subject areas) preferred to learn from examples [4]. This applies nowadays for learning the programming. That is why our students have acquired 33% of their knowledge from ready-made programs written by other programmers.

The students have mastered 69% of syntax on average, 63% of students confirmed that they would sometimes spend up to half an hour to detect common syntax errors, 78% of them agreed that the programming tools and technology should be valued based on their strengths and opportunities, as well as on the basis of their user-friendliness and ease of use.

Hypotheses

To prove the hypothesis that programming is a difficult and challenging activity, despite regular attendance at lectures, tutorials and workshops, we used a statistical method of Chi-squared test.

Method of Chi-squared test

Chi-squared test is a practical method used when one wants to determine if the frequencies obtained through the survey, deviate from the frequencies that would theoretically be expected [5].

Two groups of hypotheses

There were two groups of theses.
Group I
Hypothesis 1 (H1): Students who attend the lectures/ tutorials/ workshops regularly are able to pass the exam.
This thesis is valid if the Chi square is smaller than the critical value. In that case, there is no significant statistical difference between theoretical and empirical values, or in other words, the results from the field match the theoretical results from the questionnaire.

Hypothesis 2 (H2): Students who attend the lectures/ tutorials/ workshops regularly are still not able to pass the exam.

This thesis is valid if the Chi square is bigger than the critical value. In that case, the results from the field are significantly different from the theoretical ones, which were expected.

Table.1. Results of the survey (empirical)

	A	B	C	D
1	Achievement	Regularly attended the lectures/ tutorials/ workshops	Occasionally or rarely attended the lectures/ tutorials/ workshops	Total
2	Passed the exam	58	1	59
3	Did not pass the exam	24	5	29
4	Total	82	6	88

Table.2. Expected results (theoretical)

	A	B	C	D
1	Achievement	Regularly attended the lectures/ tutorials/ workshops	Occasionally or rarely attended the lectures/ tutorials/ workshops	Total
2	Passed the exam	54,98	4,02	59
3	Did not pass the exam	27,02	1,98	29
4	Total	82	6	88

Table.3. Calculating the values of Chi-square to check H1 and H2

	Regularly attended the lectures/ tutorials/ workshops and passed the test	Regularly attended the lectures/ tutorials/ workshops and did not pass the test
An empirical result (O)	58	24
A theoretical result (E)	54,98	27,02
Deviation (Oi-Ei)	3,02	-3,02
Square deviation $(Oi-Ei)^2$	9,12	9,12

Hi squared $(O_i\text{-}Ei)^2/Ei$	0,17	0,34

Table.4. The result of the Chi-square test method to check the H1 and H2

Result	
Chi square	0,51
Critical value	3,84
Conclusion	Hypothesis 1

Conclusion: There is no significant statistical difference between the theoretical and empirical values, i.e. the results from the field correspond to the expected theoretical results of the survey.

Hypothesis 1, which states that students are able to pass the exam if they attend the lectures, tutorials and workshops regularly, would be proven if the theoretical frequencies were exceeding 5. Unfortunately, in our analysis this was not the case, due to an unexpected high students' attendance rate. On the other hand, this may be taken as an indication that the students presumed how necessary it was to attend the educational activities regularly, what is not the case in some other courses.

Group II

Hypothesis 3 (H3): Students who attend the lectures/ tutorials/ workshops occasionally are able to pass the exam.

This thesis is valid if the Chi square is smaller than the critical value.

Hypothesis 4 (H4): Students who attend the lectures/ tutorials/ workshops occasionally are not able to pass the exam.

This thesis is valid if the Chi square is bigger than the critical value

Table.5. Calculating the values of Chi-square to check H3 and H4

	Occasionally or rarely attended the lectures/ tutorials/ workshops and failed the exam	Occasionally or rarely attended the lectures/ tutorials/ workshops and passed the exam
An empirical result (O)	5	1
A theoretical result (E)	1,98	4,02
Deviation (Oi-Ei)	3,02	-3,02
Square deviation $(Oi\text{-}Ei)^2$	9,12	9,12
Hi squared $(O_i\text{-}Ei)^2/Ei$	4,61	2,27

Table.6. The result of the Chi-square test method to check H3 and H4

Result	
Chi square	6,88
Critical value	3,84
Conclusion	Hypothesis 4

Conclusion: There is a significant statistical difference between the theoretical and empirical values, i.e. the results from the field do not correspond to the expected theoretical results of the survey.
Hypothesis 4 (H4), which states that students who occasionally attend the lectures/ tutorials/ workshops are not able to pass the exam, would be most probably proven if we had the chance to address a larger student population. However, this was not a sample but the total population available for the enquiry.
We still considered that it might be worthwhile to publish these analyses because we believe that the methodology and even the conclusions are correct, but in the concrete case, they could not meet the exact mathematical requirements to be completely persuasive.
We have also considered using the Fisher's exact test [6] but it does not seem to be appropriate, due to the relatively large size of the sample [7].

Conclusion

Programming is a very useful skill and it can contribute to a successful career. There is an ever-increasing demand for programmers, and accordingly the interest of students to master the programming is on the rise. However, programming is a challenging and difficult activity. The surveyed students who have passed the introductory course in programming had realized that. The results imply that success is possible for those who regularly attend the lectures, exercises and workshops. Of course, an additional effort at home in practicing and learning from the solved examples is required.

Programming is a skill. To become a good and versatile programmer professional in the future, our novice programmer, after his or her academic education, will have to write many lines of code. Previous practice has shown that it takes about 10 years of work and experience for a novice to become a professional programmer [8].

References

[1] Gomes, Anabela and Mendes, A. J., "Learning to program - difficulties and solutions", International Conference on Engineering Education – ICEE, Portugal (2007)
[2] Bennedsen, J. and Caspersen, M.E., "Failure Rates in Introductory Programming", SIGCSE Bull, vol. 39, 2 (2007)
[3] Hamzabegovic, J. (2014). Model of the development framework aimed to speech therapists for producing computer applications to treat dyslexia in children, (Original title in Bosnian: Model razvojnog okruženja namijenjenog logopedima za izradu primjenskih programa za tretman disleksije kod djece), Ph.D. dissertation, University of Bihać, Bosnia and Herzegovina (2014)
[4] Chi, M., Bassok, M., Lewis, M., Reimann, P. and Glaser, R., "Self-explanations: How students study and use examples in learning to solve problems", Cognitive Science, 15, pp 145-182 (1989)
[5] Shaffer, J. P., "Testing specific hypotheses in contingency tables: Chi-square partitioning and other methods. Psychological Reports", 33, pp 343-348 (1973)
[6] Fisher's exact test, https://en.wikipedia.org/wiki/Fisher%27s_exact_test
[7] http://www.mef.unizg.hr/if/alati/racunala/skripte/fisher.htm
[8] Robin, A., Rountree, J. and Rountree, N., "Learning and teaching programming: A review and discussion", Computer Science Education, 13(2), pp 137 - 172 (2003)