THE TWO WORLDS OF PROGRAMMING

SZALAYNÉ TAHY Zsuzsanna, HU
CZIRKOS Zoltan, HU

Abstract: At a faculty of a science university, like ELTE IK, programming is derived from math. At a faculty of a technical university, like BME-VIK the focus is on engineering and programming is a part of it. There are introductory curricula to programming on both faculties, but the teaching method and the paradigms being taught are different. Both faculties enrol more than five hundred students a year, but the curricula are difficult for many of them. Moreover, the number of enrolled and qualified students will be doubled in the future. Our research aims to improve the effectiveness of teaching in both faculties. Both curricula were analysed empirically based on educators' and students' opinions and observations; compared to documents, papers and exams. In order to determine the effect of prior knowledge on the results of exams, a survey was conducted in September 2015 at BME and it was repeated on the first week of the semester in 2016, at both faculties. We have selected the most determinative topics and skills that are necessary or important for passing the exams. Having compared them to the empirical analysis, we have made a proposal for the curriculum development. On the side of public education: knowing the influence of prior knowledge and the differences between curricula, students can prepare themselves more effectively for the university and by seeing the differences, they can choose the faculty which is more suitable for their personal attitude.

Keywords: programming, teaching methods, curriculum, skills

1. Introduction

Youngsters who want to become programmers have to choose a proper degree program. They find two universities in Budapest side by side where the programming courses seem to be very similar. Their typical questions are: "What are the differences between these courses?" and "Which one should I apply for?"

We explore the teaching method of basics of programming courses at Eötvös Loránd University, Faculty of Informatics (ELTE-IK) [1] and Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics (BME-VIK) [2]. They are the most important faculties in Hungary, both having 500-600 students each year. The teaching practice of programming methods is quite different at the two institutes, but the problem is the same, as everywhere in the world: learning programming is difficult, the efficiency of programming courses should be increased. Changing the research method – trying to prove a preconception – found in [3, 4], we assess wide range of prior knowledge and compare it to each result of courses to find the importance of each input skill. The published comparative analysis detects many differences in the examined courses [5]. According to the classification of [6], ELTE IK follows methodical/algorithmic- and specification-oriented method and develops a theoretical, mathematical knowledge. The Computer Science degree program focuses on programs and modelling. On the contrary, BME-VIK combines more methods as it develops practical programming knowledge. The main methods are data-, problem-, language- and instruction-oriented. The Engineering Information Technology degree program develops programming skills.

Our previous article [5] showed the differences in the curriculum but the answer to the students' second question ("Which university should I apply for?") required more research. In the second phase of our research the focus turned to skills. The questions were:

- What are the difficulties during the first semester in learning programming for students?
- What kind of prior knowledge determines the success of the course?

2. Difficulties in learning programming

The most emphasized problem at Basic Knowledge of Programming on ELTE that was mentioned by the educators is related to Nassi-Sneidermann diagrams (structograms). The curriculum describes the process of program design as follows:

1. Define requirements for input and output data using formal notation
2. Design the algorithm by drawing structograms
3. Implement the program by writing C++ code
4. Test…

As educators have reported, many students (would like to) skip the second step. Students understand what a structogram shows, they can turn it into code but they can't design it and draw it.

On the side of students, the most difficult point in the curriculum is to learn three abstraction levels in parallel. (1) Specification requires mathematical abstraction skills that are beyond the knowledge of beginners. (This is why the formal specification is usually accepted if it contains minor errors.) (2) Design requires practice in using structograms, which are easy to understand but difficult to reproduce. (3) Implementation means programming. Students prefer writing code and running the software. many of them cannot draw a structogram without trying the program first.

The Basics Knowledge of Programming course of ELTE is not only for computer scientists, but it is to be attended by informatics teachers as well. They have the same curriculum, but their results are significantly worse.

The most emphasized problem in Basics of Programming 1 at BME is using indirection, understanding the connection between the data itself and its reference (pointer). It is often seen that students' problems regarding this topic are induced by lack of understanding of earlier topics. Although they use arrays to store a series of data, indirect and random access to elements (eg. understanding the connection between elements and indices) is difficult for some. The tests at BME expect students to have practice in designing and using data structures, which is more difficult when the abstraction gets closer to the physical level.

How can these problems of both curricula be solved? Do the problems stem from the students' insufficient prior knowledge? How can educators improve the teaching effectiveness? To answer these questions a survey was conducted in September 2015 (only at BME) and it was repeated in 2016 (at both faculties), on the first week of the semester and the responses were compared to the partial and final results.

				Responses					
Group	No. of students	AVG of Marks (scale [0, 5])	DEV of Marks	No. of students	%	Freshly graduated	Graduated earlier	AVG of Marks	Dev of Marks
ELTE'16	547	3.29	1.86	284	52%	158	126	3.43	1.77
(Teacher	49	2.32	2.10	17	35%	7	10	3.00	2.06)
BME'16	617	2.94	1.76	239	39%	205	34	3.48	1.64
BME'15	565	2.88	1.76	318	56%			3.30	1.67

Table 1: The characteristics of the examined groups

3. The effect of prior knowledge on the exams

The survey contains twenty-six questions. Three of them concern the learning habits of students. Some of the remaining twenty-three ask about the achievements in secondary school subjects. Since many of the students had not taken a graduation exam in some of those subjects, the options were supplemented with an explanation of the required skills. Other questions require students to choose the best fitting answer with no scaling, but the options correspond to different levels of our Learning Activity Unit as defined earlier in [7]. In this case, the answers describe different opinions and attitudes and the assigned value shows the usefulness of the knowledge. For example, the question about the knowledge of structograms:

The structogram…
… I've never heard this word. (1)
… I can't spell it, but I've seen boxes like this stuff. (2)
… some say it is illustrative, but we should rather describe algorithms in words. (4)
… is an illustrative representation of control structures (5)

The draft analysis contains averages and deviations for all questions and for all evaluations of subjects. In order to find relations between prior knowledge and the results of evaluations, each question was correlated to every evaluation. With respect to the result of evaluations, the average value and deviation of answers is said to be high if they are higher than other values in Table 1. (This means being higher than the ½ of the cells for the averages and being higher than ¾ of the cells for the deviations.)

3.1 Observations

The most interesting finding of the survey was that the result of projects (homework) are less dependent upon the prior knowledge than any other result, in all groups.

Structograms (row 20) are little known. The correlation with ELTE'16 1st paper is much more definite than any other relation. Here another teaching method would be needed due to missing prior knowledge and its effects on the results. (The visible difference between ELTE and BME shows that knowing structograms is not required at BME.) As drawing the structogram is the problem (students cannot visualize it in advance, therefore cannot draw it), a solution would be to drop drawing and use some modelling software instead. The order of papers could also be changed to give students more time to practice. Practically this means that educators should accept if students prefer writing code first. Also, educators should realize that structograms do not aid the designing of algorithms. It is only a form of documentation.

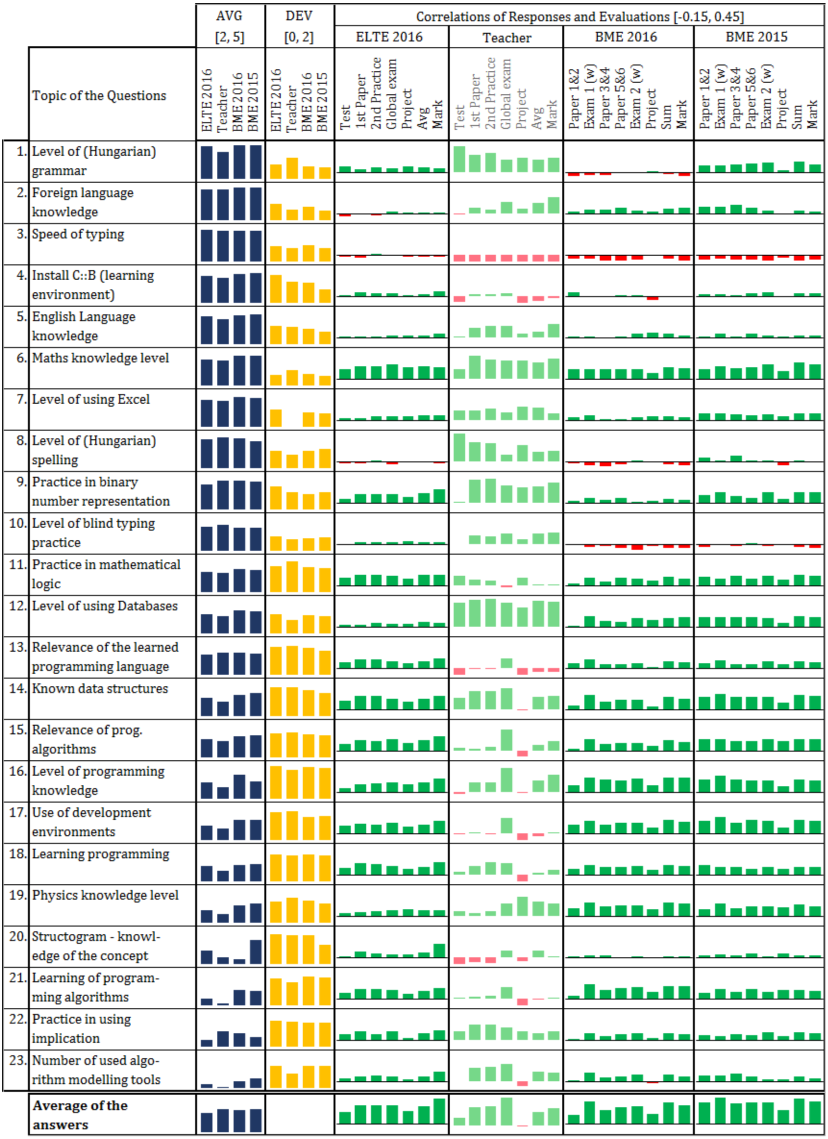

Table 2: Input Survey – Result and Correlation to Qualification

Database knowledge (row 12) is more relevant at BME than ELTE (correlations are higher). The lack of prior knowledge could cause problems, as mentioned previously. The high level of spreadsheet knowledge reported by the students could be the basis of introduction to references and indirection. Although teachers form a very small group at ELTE, all correlations to database knowledge are three times higher than in other groups.

The only mandatory subject for university admission is Math. It features high average and low deviation and it is very determinative (row 6). Some topics of Math were specifically in the questionnaire (rows 9, 22). The comparison of little deviation of Math grades and the deviation of results in programming courses shows that even a small deficiency in Math knowledge is reflected in drastically worse results.

On the other hand, Physics (row 19) has low average and shows high deviation. The correlations show that the importance of Physics on BME is higher than on ELTE.

The highest correlations with ELTE's and BME's evaluations are seen in row 13-18. The averages and deviations of these questions are close to the final evaluation. All six questions focus on programming skills. The aim of question 16 was to determine the relevance of learning programming in high school. The result shows that basic programming skills are very important for success. This prior knowledge, programming itself, is the foundation for development. Regarding the teachers, only the final exam has high correlation with this question, and the results are poor.

An interesting question arises when exploring the answers regarding typing, foreign language, spelling and grammar (rows 1, 2, 3, 5, 8, 10). These answers give very high averages and close to zero, sometimes negative correlation to evaluations, except for teachers. Deviations are not reflected in evaluations. The result shows that if the programming skills are high enough, the average of prior knowledge in linguistic topics is higher than needed. Regarding these topics, the less prepared students are as successful as the most prepared ones. Negative values on correlations show that written tests could be more difficult for a perfect typewriter because she does not practice handwriting. Moreover, students who have difficulties in these topics are helped in administrative way (they get more time). But in case of missing programming skills, these skills can become advantages.

4 Proposals, considerations

4.1 Students' aspects

Students are suggested to be as good as possible in topics where the correlation and the deviation are high, because those topics seem to be determined by prior knowledge. The most important topics that have the strongest correlation with the tests, in 2016 on ELTE are:

1. Maths knowledge level
2. Known data structures
3. Relevance of programming algorithms
4. Practice in mathematical logic
5. Use of development environments
6. Learning programming

The 1st, 3rd and 4th topics form the basis of the curriculum. It should be clear that inside the general Math knowledge, mathematical logic is emphasized. The focus of the curriculum is on algorithms but there are no programs without data. Also, the development environment is out of teaching focus. Therefore, without knowing the basics of programming, studies become difficult.

The topics with the highest correlations on BME in 2016 are:

1. Level of programming knowledge
2. Physics knowledge level
3. Use of development environments
4. Learning of programming algorithms
5. Maths knowledge level
6. Known data structures

The same topics are seen in the previous year's research in a slightly different ordering. The most important requirement here is to have practice in programming, some algorithmic skills and some familiarity with development environments. This is the foundation of learning dynamic data structures and C language. Physics is as important as Maths, but less important than on ELTE.

Comparison of two training courses: ELTE expects precisely planned algorithms that obey the rules of structural programming and use complex logic expressions while BME uses a more permissive and practical approach and let students to use break statement and multiple function exit-points.

On the other hand, BME expects precisely defined data structures and dynamic memory allocation, while the curriculum of ELTE is more relaxed regarding this topic, suggests estimating upper bounds for array sizes. The two different approaches are stated clearly in priorities, expectations and everyday teaching practises.

4.2 Curriculum development

Educators are suggested to change those teaching methods where students' result is low but the correlation with the prior knowledge is high. This combination shows that the students' results come from the prior knowledge rather than effective learning during the course. Exploring the averages, we can see the yearly changes in knowledge of enrolled students. Table 2 implies that structograms and other algorithm modelling tools are less and less taught in public education.

Additional results can be found when we explore the differences between the group of Informatics teachers and engineering/computer science students. In teachers' education, lower level of prior knowledge in programming is seen. Students are not prepared well enough, yet they seem to overestimate their Maths knowledge. Since the group is small (17 people), further research is needed. Their results are so different that it raises the question whether they need a different learning method than computer scientists. They are not required to design complex programs, nor use mathematical representations.

Summary

Programming curricula at science and technical universities are like yin and yang. Two opposite parts that complement each other. The input survey presented in this article covers both parts and compared the responses to the results of the evaluations. The analysis shows the current status and can help to determine the areas of development of teaching and learning methods to reach a valuable, high average output. The input survey, compared to earlier and other groups' results, can help to adapt the prior knowledge of students and choose effective teaching methods. Analysing the prior knowledge of enrolled students and the efficiency of course in informatics teacher education would be more important. It seems to be another world of programming.

Bibliography

1 Papp, G., Horváth, G., Szlávi, P. and Zsakó, L.: *Programozási alapismeretek*. ELTE IK, Budapest, 2016. [Online]: http://progalap.inf.elte.hu/ [Accessed on: 28 April 2017]

2 Czirkos, Z.: *INFOC*. BME EET, Budapest, 2016. [Online]: https://infoc.eet.bme.hu/ [Accessed on: 28 April 2017]

3 Dehnadi, S., Bornat, R., Simon: Mental models, consistency and programming aptitude. In: *Proceeding ACE '08 Proceedings of the tenth conference on Australasian computing education* - vol 78 p 53-61 ISBN: 978-1-920682-59-0

4 U. Nikula, O. Gotel és J. Kasurinen, „A Motivation Guided Holistic Rehabilitation of the First Programming Course, In: *ACM Transactions on Computing Education*, vol 11, No 4, p 24:1--24:38. 2011. DOI: 10.1145/2048931.2048935

5 Szalayné Tahy, Zs. and Czirkos, Z.: "ProgAlap" és ami mögötte van. *8. InfoDIDACT*, Zamárdi: Webdidaktika Alapítvány, 2015. ISBN: 978 963123892 1

6 Szlávi, P. and Zsakó, L.: Methods of teaching programming. In: *Teaching mathematics and Computer Science,* vol. 01(2), pp. 247–258., 2003. DOI: 10.5485/TMCS.2003

7 Szalayné Tahy Z., Czirkos Z.: "Why Can't I Learn Programming?" The Learning and Teaching Environment of Programming. In: *Brodnik A., Tort F. (eds) Informatics in Schools: Improvement of Informatics Knowledge and Perception*. ISSEP 2016. Lecture Notes in Computer Science, vol 9973. Springer, Cham DOI: 10.1007/978-3-319-46747-4_17

Reviewed by: Sándor Király PhD, associate professor

Contact address

Zsuzsanna SZALAYNÉ TAHY, HU

Eötvös Loránd University, Faculty of Informatics,

Pázmány Péter sétány 1/c, Budapest 1117. Hungary

e-mail: sztzs@caesar.elte.hu

and

Zoltan CZIRKOS, HU

Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics

H-1117 Budapest, Magyar tudósok körútja 2.

e-mail: czirkos@eet.bme.hu