Original Article

IJEEE

International Journal of Electrical Engineering
Education
0(0) 1–10
© The Author(s) 2018
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0020720918773975
journals.sagepub.com/home/ije
SAGE

Learning a programming language

Olivera Iskrenovic-Momcilovic

Abstract

Today, it is believed that it is difficult to learn a programming language because it requires a completely different way of thinking from what students are accustomed to and need a lot of work and time to start thinking in this way. Each method for learning a programming language aims to improve the quality of learning, which is confirmed by a large number of successful students. This paper presents the influence of factors such as motivation, emotion and pre-knowledge of students on the choice of method, which is maximally adapted to the age, abilities and preferences of students. Adequate selection of the method is done by using the tests for checking the given factors to the students' data, which are later used to select the most favorable method for learning a programming language.

Keywords

Learning, programming language, pre-knowledge, motivation, emotions

Introduction

Teachers, lecturers and researchers have long been interested in researching factors that contribute to increasing student learning efficiency. These factors can be within or outside the school.[1] According to Mushtag and Khan,[2] these factors can be classified as social, psychological, economic, personal and environmental. However, getting to know each student as an individual most helps to understand which factors can influence his or her process of learning and memorizing

Faculty of Education, University of Novi Sad, Sombor, Serbia

Corresponding author:
Olivera Iskrenovic-Momcilovic, Faculty of Education, University of Novi Sad, Sombor, Serbia.
Email: oljkaisk@yahoo.com

materials. The most common factors that can influence the students' success when learning new materials are pre-knowledge, motivation and emotions.

Pre-knowledge

According to Piaget,[3] children during cognitive development change the way in which they establish contact with their environment and the way they perceive the world around them. Piaget considered that children were "small researchers" who actively sought to explore and understand the world around him. By the moment they enroll in school, they have already developed informal theories about world, themselves and others. Learning depends largely on what this person knows before, so acquiring new knowledge takes the form of an upgrade to already existing knowledge and experience. All this makes it necessary to distinguish the terms "pre-knowledge" and "knowledge." Marzano[4] simply describes knowledge as what a person already knows about content, while Biemans and Simons[5] define pre-knowledge as "all the knowledge that students have when entering a learning environment, which is potentially important for the acquisition of new knowledge." Dochy and Alexander[6] provide a much more elaborate definition describing pre-knowledge, metacognitive knowledge and conceptual knowledge.

Many papers suggest that teaching of should be built on the student's pre-knowledge.[7–9] The majority of the students know informal methods from everyday life, which resemble learning a programming language. Younger students know that they compare their mates to the height, and older students to select the book according to the given criteria. Such knowledge can be acquired when the basic commands of the programming languages are learned. Teachers can ease the learning process by helping students to activate pre-knowledge.[10] However, when students create a series of wrong estimates, foreground activation can disrupt the learning of new content. Pre-conceived ideas can be interfered with or mixed with new content. In order to resist the potential negative impacts of pre-knowledge, teachers and students should set aside time and effort to make thinking better and more flexible.[11]

Even a superficial amount of correct knowledge in a short period of time greatly improves the learning process and allows for its proper expansion in the future.[4] It would be wrong to think that the influence of pre-knowledge on learning is negative. This is not the case here. Learning first of all begins with the familiar and extends into the unknown. Linking day-to-day teaching experiences and deliberately fostering previous knowledge with new teaching contents can provide meaningful and long-lasting knowledge.

Motivation

Learning a programming language is in itself a problem because it is a skill that is difficult to learn. To successfully access the programming skill, students need to be motivated.[12] Motivation can be viewed on two sides: as a driver of a student or a teacher for work.[13] Motivation is a very abstract concept for which there is no real measure.[14]

There are many prejudices and stereotypes in the area of learning a programming language. Students who start education are highly motivated for every type of learning. They are full of life and interest, because they go into something new, unknown. However, over time, that serenity is falling and teachers sometimes do not understand that everything has gone wrong.[15] There are several factors that lead to the loss of interest in learning a programming language.

It is interesting to note that students consider the introductory lessons uninteresting for the tasks presented to them, such as, for example "Sort list" or "sum up all integers to n".[16] Learning a programming language is not just "learning of words," it is one very interesting area. One of the ways that can draw us into programmed computer games. It is believed that today's and new generations will be more interested and motivated by the approach, which uses computer games than traditional introductory lessons of programming language.[16]

Some believe that learning a programming language is difficult because it is associated with high intelligence and logical thinking. Others think that programmers lack interpersonal skills, feel social discomfort and are obsessed with technology because they have no other interests. All this can discourage an individual to start learning a programming language or to accept programming as a necessary skill for his or her professional development. These negative attitudes can change when an individual experiences his or her own experience of learning a programming language.[17] If teachers have positive attitudes toward programming languages, then they will affect the motivation of their students.[17]

Emotions

In many studies, it has been shown that there is a significant correlation between mood and cognitive processes.[18] A positive mood is a mental state in which a person feels more confident, more optimistic, and more freer than usual. Individuals who have positive moods process information less systematically, and more creatively and flexibly than those who have negative moods.[19] When a person feels anxiety, depression or fatigue, it can be said that they are in a negative mood. A negative mood sometimes affects the data processing in a more systematic and less creative way. A negative mood worsens the learning process and prevents academic success.[19]

Today, the influence of emotions on students is intensively investigated during their first encounter with learning a programming language.[20] The ideal emotional state of learning is where students are most capable of acquiring new information.[21] However, emotions can increase or decrease learning outcomes. For example, confusion and frustration are expected to emerge rapidly when the results of the program do not match expectations (confusion) or the student has no idea how to continue the program development (frustration). Failure in program design is always associated with frustration and lower efficiency, which can lead to boredom and lack of interest.[22] Therefore, the general hypothesis is that emotions play an important role in the process of learning a programming language and can also affect both direct, such as failing exam, and long-term outcomes, such as dropping out of the programming course.

Emotions have an impact on the debugging and coding skills of the programmer. However, there are few studies that examine the importance of emotions in software engineering. Some researchers believe that developers are much different from other people. They are experiencing frustration during the development of the program, which reduces productivity, but also anger, which results in increased productivity.[23] On the other hand, the enthusiasm state was assessed as having the most positive impact on productivity. There is a completely similar situation in the field of debugging. All the induction, closure, memory, reasoning procedures associated with debugging are cognitive, indicating the impact of emotion on the elimination of errors.[24] To date, there is little research in the field of debugging and cognitive abilities.

Methodology of research

The aim of the research is to create a model that, based on three parameters pre-knowledge, motivation and emotional state, selects the method for the learning a programming language, which is best adapted to the student and his or her abilities. The methods used to teach students are:

1. descriptive method, using sentences
2. graphic method, using a flow diagram
3. pseudocode, using a language that is close to spoken language
4. program code, using a strictly formalized programming language.

Methods have been sorted out so that the student, who is least interested in programming, gets not the shortest, but certainly the most understandable method, which resembles the spoken language. It starts from the descriptive method for the weakest level, through the graphical method for slightly heavier pseudocode levels for the middle level to the program code for the highest level. The program code is interesting but at least understandable, so it is intended for students who have a lot of knowledge, are motivated and disposed to learn programming languages. In this case, long explanations and great simplifications reduce interest, motivation and emotions.

The tasks that are solved during learning a programming language are adapted to the chosen method of work. Students, who use the descriptive method, work the easiest tasks and have many additional and detailed explanations. In this case, they often repeat previously acquired knowledge. Students who use the graphic method work harder tasks, but they have fewer additional explanations. Finally, in the case of pseudocode and program code, students work hardest tasks and have at least additional explanations.

The sample consists of 28 students of the third grade of the gymnasium, who attend the subject Computing and Informatics. To determine the method that best corresponds to one student's abilities, there are three different tests:

- the basics of programming test for the age of the third grade of gymnasium,
- the motivation test for the subject Computing and Informatics,
- the test of the emotional state of the students during the examination.

The basics of programming test are of informative character and serve to assess student's pre-knowledge, where the marks are: insufficient (1), sufficient (2), good (3), very good (4) and excellent (5).

In order to measure the motivation of students, we used the motivated strategies for learning questionnaire (MSLQ) developed by Pintrich et al.,[25] a widely used instrument for investigating self-regulated learning strategies. The MSLQ uses 44 5-point Likert scale questions (1 = strongly disagree, to 5 = strongly agree) to measure students' motivational orientations. Since self-regulated learning is context specific[25] and our purpose is to measure motivation in the specific context of learning a programming language, the survey questionnaire is directly defined in the context of specific subject of programming. For example, a self-efficacy item is "I believe that I am able to learn well how to program"; an intrinsic motivation item is "Even when completing the assignments does not guarante that I get a good grade, I still love to complete them." If the student has the most answers "strongly agree," his or her motivation for learning a programming language is excellent (5), in the other words if he has the most answers "strongly disagree," his or her motivation is insufficient (1).

The achievement emotions questionnaire (AEQ) is a multidimensional self-report instrument developed to measure the emotions of students in academic situations.[26] There are three sections to the AEQ, containing the class-related, learning-related, and test-related emotion scales. For the present study, we used the third section, that consists of 77 items, which must be answered on the basis of a 5-point Likert scale (1 = strongly disagree, to 5 = strongly agree). If the student has the most answers "strongly agree," his or her emotion is enjoyment (2), then the order of happiness (1), neutral (0), anger (−1) and boredom (−2).

On the basis of the tests and their results, the criteria for choosing an appropriate method for learning a programming language are obtained. The selection of an adequate method is carried out using the network method of multi-criteria analysis. Starting from the values of the following two parameters:

- pre-knowledge – insufficient (1), sufficient (2), good (3), very good (4) and excellent (5)

- motivation – insufficient (1), sufficient (2), good (3), very good (4) and excellent (5)

For easier orientation, the following tags were introduced:

- Pi – pre-knowledge with values: 1, 2, 3, 4 and 5,
- Mi – motivation with values: 1, 2, 3, 4 and 5,
- PiMi – combination pre-knowledge and emotion, with values: 1, 2, 3, 4 and 5.

Of all values of Pi and Mi, 25 combinations of PiMi can be made, each of which has its own value. Each value PiMi is calculated as the integer mean value of the values of Pi and Mi, because it has the same effect of pre-knowledge and motivation on learning. The grouping of combinations was made to simplify future considerations. All values of PiMi are grouped into five categories Li according to their values: 1, 2, 3, 4 and 5. It should be noted that this form of grouping is selected because it is the simplest.

After grouping, the influence of the emotions values Ei is considered. The choice of the emotions values was done in the following way:

- The emotions of rage and boredom are negative emotional states that negatively affect the cognitive abilities of the student, so their values are -1 and -2,
- Neutral emotional state has no influence on students' cognitive abilities, so its value is 0.
- Emotions of happiness and enjoyment are positive emotional states, which improve the cognitive abilities of student, so their values are 1 and 2.

Of all values of Li and Ei, 25 combinations LiEi can be made, each of which has its own value. Each value LiEi is calculated as the integer average value of Li and Ei, because it has the same effect of pre-knowledge, motivation and emotion on learning. It should be noted that in order to avoid negative values, we define a rule if the value obtained is less than one, the value LiEi is one. For better transparency, the LiEi values are grouped into four categories LLi, according to their values: 1, 2, 3 and 4. It should be noted that this form of grouping is selected because it is the simplest. Categories LLi represent methods for learning a programming language.

Results and discussion

Tests for pre-knowledge, motivation and emotional state were done by 28 students of the third grade of the gymnasium. From Table 1, it can be seen that four students had insufficient, seven enough, four good, six very good and seven excellent knowledge in basics of programming. Most of them were highly motivated (10 students), followed by very good (7 students) and good (5 students) motivation, and ultimately the equal number of those with sufficient (3 students) and weak

Table 1. Results of tests for students' pre-knowledge, motivation and emotion.

Pre-knowledge		Motivation		Emotions	
Marks	Students	Marks	Students	Marks	Students
Excellent (5)	7	Excellent (5)	10	Enjoyment (2)	8
Very good (4)	6	Very good (4)	7	happiness (1)	9
Good (3)	4	Good (3)	5	Neutral (0)	7
Sufficient (2)	7	Sufficient (2)	3	Anger (−1)	3
Insufficient (1)	4	Insufficient (1)	3	Boredom (−2)	1

(3 students) motivation for programming. Based on the monitoring of emotions, it was found that the highest number of students in the test was in positive emotional state (enjoyment – 8 and happiness – 9 students), seven students in neutral emotional state and least in negative emotional state (anger – 3 and boredom – 1 student).

Based on the results of the tests, according to the methodology described for each student, the method for learning a programming language is selected, which suits him best. The student who achieved poor performance with insufficient knowledge, low motivation, and whose emotional state was bored has been given the easiest method – the descriptive method, which provides enough clues for the student to spread his or her little knowledge in an easy and interesting way but also to adopt new material. This method uses a spoken language to explain in an easy and understandable manner any command of the programming language. It is good for beginners because they have shown better results than on the basics of programming test.

The student, who achieved a little better on tests, was assigned a second weight method – the graphical method. It is based on drawing symbols associated with arrows (external paths), which define the course and direction of the program. The graphic display is simple, transparent, easy to find errors. In addition, the problem can be easily analyzed, compared to some other problem, and thus shorten the time for finding a solution. The student considers this method interesting and unusual, he wants to learn and gain motivation for pedagogy. As he is positively excited about what is good for him, his emotional state is lifted, giving him better concentration and, therefore, better results.

Students who had medium success on the tests received a third-weight method, i.e. a pseudocode, which is implemented in a language that is close to them. It is not written in any of the programming languages, but with agreed words and characters from the spoken language. So, we need an agreement, which we will use words and signs. A considerable level of knowledge and motivation is needed in order for the steps to be followed and not to be confused. Nevertheless, this is certainly an interesting method that gives students the opportunity to learn through creativity and raise their motivation for learning a programming language and bring them into a positive emotional state. The students who used this method achieved good results and were very interested in this method.

Students with the best results in the tests received the most difficult method, using the classic technique of writing a program in a programming language. This method is most commonly used in regular classes and is considered less interesting by students. Since a student who receives this method already has enough interest in learning a programming language, as well as an excellent knowledge, he does not need any additional incentive for unusual and interesting methods. The students who used this method have achieved solid results, which are expected from them, as they did not have problems on the tests. Some achieved slightly poorer results due to the influence of inadequate emotional state (fear), and they were given a new method, which resulted in better success.

Conclusion

Each learning method aims to improve the quality of learning. However, some methods cannot achieve the desired results because they are not adapted to the learner as an individual. Individual student characteristics are a key reason why some students find it difficult to master teaching materials and have poor results on tests in some subjects. In order for students to accept new knowledge, they should be allowed to learn the way they can and they want.

Today, students have the most problems while learning a programming language. Teachers keep classes that are boring, monotonous and unmanageable to the individual needs of students. This type of teaching material, which predominantly uses chalk and board, significantly reduces the motivation and emotional state of the students. The only solution is that teaching materials and teacher exposure match the individual characteristics of students. Students should consider learning a programming language as one positive experience, which can be useful in a future career or in everyday life.

This paper examines the influence of factors such as: pre-knowledge, motivation and emotional state of the students. Firstly, data on these factors are based on the tests, and then the choice of the method of learning a programming language is made, which is maximally adapted to the age, abilities and preferences of the students. In this way, students, who are without adequate knowledge, are not motivated and unwilling to learn, can become motivated and have the will to learn. The student rather accepts a method that is adapted to him than a standard method. Therefore, the teacher should choose the method of learning, which would bring the student closer to the teaching material, facilitate learning and improve school success.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Olivera Iskrenovic-Momcilovic http://orcid.org/0000-0003-0162-4416

References

1. Farooq MS, Chaudhry AH, Shafiq M, et al. Factors affecting students' quality of academic performance: a case of secondary school level. *J Qual Technol Manage* 2011; 7: 1–14.
2. Mushtag I and Khan SN. Factors affecting students' academic performance. *Global J Manage Business Res* 2012; 12: 17–22.
3. Piaget J. *On the development of memory and identity*. Barre, MA, USA: Clark University Press with Barre Publishers, 1968.
4. Marzano RJ. *Building academic background knowledge*. Alexandria, VA, USA: Association for Supervision and Curriculum Development, 2004.
5. Biemans HJ and Simons PR. CONTACT-2: a computer-assisted instructional strategy for promoting conceptual change. *Instr Sci* 1996; 24: 157–176.
6. Dochy FJ and Aleksander PA. Mapping prior knowledge: a framework for discussion among researchers. *Eur J Psychol Educ* 1995; 10: 225–242.
7. Costley KC, West HG. Teaching practice: a perspective on inter-text and prior knowledge. *SRATE J* 2012; 21: 21–25.
8. Dunlosky J, Rawson KA, Marsh EJ, et al. Improving students' learning with effective learning techniques: promising directions from cognitive and educational psychology. *Psychol Sci Pub Int* 2013; 14: 4–8.
9. Kudryashova A, Gorbatova T, Rybushkina S, et al. Teacher's roles to facilitate active learning. *Medit J Soc Sci* 2016; 7: 460–466.
10. Strangman N, Hall T and Meyer A. *Background knowledge with UDL*. Wakefield, MA, USA: National Center on Accessing the General Curriculum, 2004.
11. Campbell L and Campbell B. Beginning with what students know – the role of prior knowledge in learning. In: Campbell L and Campbell B (eds) *Mindful learning: 101 proven strategies for student and teacher success*. Thousand Oaks, CA, USA: Corwin Press, 2009, pp.7–21.
12. Feldgen M and Clua O. Games as a motivation for freshman students to learn programming. In: *34th ASEE/IEEE frontiers in education conference*, Savannah, Georgia, USA, 20-23 October 2004.
13. Jenkins T. The motivation of students of programming. *SIGCSE Bull* 2001; 33: 53–56.
14. Entwisle N. Motivation and approaches to learning: motivating and conceptions of teaching. In: Brown S, Thompson G and Armstrong S (eds) *Motivating students*. London, UK: Kogan Page, 1998, pp.15–24.
15. Zainal NFA, Shahrani S, Yatim NFM, et al. Students' perception and motivation towards programming. *Procedia* 2012; 59: 277–286.
16. Combefis S, Beresnvicius G and Dagiene V. Learning programming through games and contests: overview, characterisation and discussion. *Olymp Inform* 2016; 10: 39–60.

17. Charters P, Lee MJ, Ko AJ, et al. Challenging stereotypes and changing attitudes: the effect of a brief programming encounter on adults' attitudes toward programming. In: *Proceedings of the 45th ACM technical symposium on computer science education*, Atlanta, Georgia, USA, 5-8 March 2014.
18. Graziotin D, Wang X and Abrahamsson P. How do you feel, developer? An explanatory theory of the impact of affects on programming performance. *Peer J Comput Sci* 2015; 1: e18
19. Febrilia I and Warokka A. The effects of positive and negative mood on university students' learning and academic performance: evidence from Indonesia. In: *Proceedings of the 3rd international conference on humanities and social sciences*, Hat Yai, Songkhia, Thailand, 2 April 2011, pp. 1–12. Thailand: Faculty of Liberal Arts, Prince of Songkla University.
20. Bosch N, D'Mello S and Mills C. What emotions do novices experience during their first computer programming learning session? In: *Proceedings of international conference on artificial intelligence in education*, Memphis, TN, USA, 9-13 July 2013.
21. Lee DMC, Rodrigo MMT, Baker RSJ, et al. Exploring the relationship between novice programmer confusion and achievement. In: D'Mello S, Graesser A, Schuller B, et al. (eds) *Affective computing and intelligent interaction*. Springer: Berlin Heidelberg, pp.175–184.
22. D'Mello S and Graesser A. Dynamics of affective states during complex learning. *Learn Instruct* 2012; 22: 145–157.
23. Wróbel M. Emotions in the software development process. In: *Proceedings of 6th international conference on human systems interaction*, Gdansk, Poland, 6-8 June 2013.
24. Khan IA, Brinkma WP and Hierons RM. (o moods affect programmers' debug performance? *Cogn Tech Work* 2013; 13: 245–250.
25. Pintrich PR, Smith DA, Garcia T, et al. *A manual for the use of the motivated strategies for learning questionnaire (MSLQ)*. Ann Arbor, USA: National Center for Research to Improve Postsecondary Teaching and Learning, University of Michigan, 1991.
26. Pekrun R. The control-value theory of achievement emotions: assumptions, corollaries, and implications for educational research and practice. *Educ Psychol Rev* 2006; 18: 315–341.