



ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2019; 5(8): 146-149
www.allresearchjournal.com
Received: 10-06-2019
Accepted: 13-07-2019

Massoud Massoudi

1. Assistant Professor, Faculty of Computer Science, Parwan University, Afghanistan
2. Ph.D. Scholar at CSE, Department of Delhi Technological University, New Delhi, India

A review on challenges and solutions in learning programming courses at undergraduate level

Massoud Massoudi

Abstract

Taking a computer programming language course for the first time is a problematic and challenging task for most of the pupils who mostly fail or drop out of the course. In addition to this, this problem is not restricted for computer science pupils, since pupils in other engineering disciplines have to take programming courses as well.

In this paper, we review papers related to teaching approach, challenges and solutions for improving student's ability and performance in a first programming language course in undergraduate program.

One of the most well-known challenges in teaching and learning a first programming language course is that in the same course, the students must learn and get to know fundamental programming techniques, high level abstraction capabilities, the application of those tricks and techniques, notion and significance in problem solving and design.

In this paper the general trends comparing freshers and expert programmers, programming strategies and knowledge, program generation and understanding has been identified.

The main and fundamental focus and point of this review is on novice programming and concepts relating to novice learning and teaching. Different problems faced and experienced by novices are recognized, containing issues relating to core program design, to algorithmic complexity in some specific language features, to the frangibility of novice knowledge, and so on. Keep this in mind we have summarized this material and suggest some practical concepts for academicians.

Keywords: Computer programming language undergraduate

1. Introduction

The far-reaching use of programming languages commenced with the arrival of Fortran in 1957. It gave the opportunity for scientist and engineers to write different formulas using standard symbols from mathematics [8].

Coding and programming is the most interesting skill and most of the developers consider this skill as fun and can be a remunerate career. In recent years the requirement and demand for programmers, coders and pupil's interest in programming have grown quickly and rapidly, on the other hand, the introductory programming courses have become increasingly reputed. However, Learning to code and program is seems to be hard and challenging. Novice programmers mostly suffer from a vast range of difficulties and deficits. Programming subjects are mostly considered as challenging, and often have the highest drop-out rates. In general it is accepted that it takes approximately 10 years of experience to change and convert a novice programmer into an expert one [1].

The high omission and drop-out rate and failure measure in programming courses have drawn serious attention of researchers to start investigation of the causes and solutions to the drop-outs. Programming and coding is a multi-layers skill which is boring, scary and unrelated to day-to-day experiment where pupils only can learned in single context [2].

Our interest in this broad field of software engineering is focused by practical sessions and considerations. In [3] have stated that we teach the introductory programming course named as CS1, where our objective is to provide the best teaching and learning environment and experience in doing practical for our pupils. In the result, we are more eager in understanding the processes of teaching and learning programming, and come this point, Why is programming is difficult to learn? In the same [3] has been mentioned, that the only effective approach to learn a new programming language is by writing and coding the programs in it. Where the first program for all the languages is same such as print the "Hello, World"

Correspondence

Massoud Massoudi

1. Assistant Professor, Faculty of Computer Science, Parwan University, Afghanistan
2. Ph.D. Scholar at CSE, Department of Delhi Technological University, New Delhi, India

this is the big obstacle, to bounce over it you have to be able to create the program source code somewhere in a text editor or IDEs, compile it without generating any error, run it and find out what output you will get and where your output went ^[3].

This paper presents a review on the challenges and difficulties where undergraduates encountered while learning programming languages and their perspectives on how learning should naturally take place. The rest of the paper is organized as follows: section II Teaching & Learning Programming Languages and Environment, III Languages Readability and Writability, IV the Role of Programming Languages, V Challenges in Learning Programming Language, VI Solutions, VII Objective Measures and Test and the conclusion is stated in section VIII.

2. Teaching & learning programming language and environments

Knowing and Understanding the learning process of any first programming language can cause an effective learning environments as much as you learn and code you will create your own appropriate learning environment. Some lecturers and systems consider the programming in an entertaining way like game playing, storytelling and fun ^[4-5].

Some of the most potential list of programming language advantages are state as:

- It Increased capability to express your ideas in better way, it is widely accepted that as much as at depth we can think and use our mind it causes to express the power of the language by which we communicate our thoughts and ideas.
- It can improve background for selecting appropriate languages, most of the expert programmers have had little formal education and degree in computer science, hence, they have learned programming on their own or through in-house training courses and institutions, the more you program the more you will get to know which language can help me in developing most needed applications.
- It can enlarge ability to learn new programming languages. Computer programming language is a young discipline and design methodologies, programming languages and software development tools are still trending, where this makes the software development process an exciting and interesting profession.
- Better understanding of the importance of implementation. When learning the concepts of any programming languages it is both interesting and essential to touch on the implementation issues that affect the concept. However, in some cases an understanding of implementation issues leads to an understanding of why programming languages are designed the way they are. This in turn leads to the capability to use a language more efficiently.
- It can improve the ability to design new Language. For most of the students the requirements in the future to design their own programming language seems like remote and impossible. However, most of the programming languages have been occasionally design in different sort by professional and expert programmers.
- Overall improvement in computing. As a final point, a global opinion of computing is require to rectify the study and learning of programming concepts and that is

there. Although, a question may raise in your mind to determine why and how a particular language became popular and famous, this is not always vivid it is usually possible to determine why a particular programming language became popular, it is not always clear, at least in retrospect, the most popular and used programming languages are the one which are best available.

In general, if those who select languages are better informed, perhaps, that better programming languages can quickly squeeze out the poorer ones ^[6].

3. Languages readability and writability

Programming language is a digital device language designed to associate and communicate different instructions to a computer and device. They are used to create different programs and applications to control the behavior of a computer as well as to express algorithms. There are so many programming languages which all can do the same task but in different ways using different syntaxes. Hence, Most of the programming languages characterize computation in an imperative format and style, i.e., as a sequence of instructions and commands, however, some programming languages that support functional programming or logic programming use alternative forms of description ^[17].

One of the most significant criteria to judge a programming language is its simplicity and its ease, where the program can be read and understood easily, on the other hand, readability has to be considered in the context of the problem domain.

Whereas, Writability is considered as a measure of how simply a language is used to create programs for a specific problem domain. As a result, Most of the languages that affect the concept of readability also affect Writability of that language.

In view of the fact, both concept can influence reliability ^[6]. In addition to this, most of the developers who spend majority of their time trying to read and understand the addressed source code during the maintenance phase. This results that spending less time on reading the source code has direct impact on the amount of time for maintaining the software.

As a final point, the concept of Code readability is highly subjective and it is directly related to the level of simplicity by which a developer reads the program. It should not be unsaid, that Readers' levels mainly depend on their comprehensive and cognitive ability to process textual information ^[7].

4. The Role of Programming Languages

To know more the role of any programming languages, let's consider the below questions that results in recognizing the role of languages in our daily development domain.

Is a programming language a tool for guiding and instructing systems? Is it a means for communicating between programmers? Is it a device for expressing high-level designs? Is it a way of expressing relationships between concepts? Or is it a means for controlling digital devices? In My perspective a general-purpose programming language must be all of those to serve its diverse group of users. Programming languages are instructions, they are used for organizing, specifying and reasoning about computations ^[8].

In this digital era, it is required to everyone should learn how to program and how to code! Hence, Programming is not only designed to a specific discipline limited to computer science students. Programming is a good way of thinking where it is useful to everyone who uses digital devices for their daily communications. On the other side, Programming teaches most of the same skills as mathematics and English. Each Programs have to be designed with precision. Reading and coding them has to be done with caution. But the reward for a correct program is immediate and strong: it executes correctly and gives required and expected output and result ^[9].

Since the 1950s, grading and scores of different programming languages have been implemented and designed and many more will come. Those with a relevant experience in this field will agree on that concept of “the best programming language” is something meaningless. In addition to this, we need to clarify this to our pupils during teaching programming languages. In fact, some languages are better than others based on some specific purpose and indeed: (a) some languages are exclusively used for some specific purposes, and (b) people use some specific languages for specific purposes. For example, C language ^[7-5] is a good language for developing system software’s and programs. It is a low-level and not object oriented language, as well as it is very fast language and gives programmers the opportunity to directly interact with the hardware, but it is not the best language for numerical and scientific computing now a days. In view of the fact I it is believed and accepted that C language is an inappropriate language and it is not fair to teach it as “Introduction to Programming” course. That is, because some of its characteristics are so low-level it tends to focus more on the hardware, instead of on the algorithms ^[10].

5. Challenges in learning programming languages

Teaching courses which have practical aspects have always formed and constituted burdens on colleges offering non-technical degrees. One of such courses is computer programming languages, especially in degrees such as: information systems, computer science, and software engineering. In such courses, pupils are expected to take between 2-5 computer programming courses during their undergraduate level. Studies show that success ratio in the first programming course is mostly low. Why is it low and how this ratio can be improved is stated in this paper ^[11].

Programming courses are among the main components of the curriculum to be observed, in most of the fields including mathematics, Science and Engineering at tertiary levels. But not only in the field of Information Technology. In view of the fact, the course is considered difficult, challenging and it is also categorized as part of the seven grand challenges in computing education ^[12].

Different research have stated regarding the struggling of the students with first steps of learning to program and code. A multi-national, multi-institutional study was conducted to dig and explore pupils’ programming capabilities and skills and ^[13, 14, 15]. The study disclosed that pupils have mostly problems in coding ^[13], reading ^[14] as well as designing ^[15] a simple programming code.

Accordingly, it is necessary to interrogate the factors that guide to pupils learning difficulty in programming. The difficulties need to be rectified in order to be able to aid them learning in an effective and suitable way.

6. Solutions

Every problem comes with so many solutions. Hence, A helpful approach in learning how to code requires students to consider a real world problem and then converted it to program code to solve it. However, such methods has been difficult by students because the learning process of programming is said to be a multi-layers skill which more boring, intimidating and as well as not related to day-to-day experience where pupils mostly learned in single context ^[13]. Such situation often happened to a novice programming student with no experience in this field where at the starting stage they need to learn specific programming language syntax before applying their knowledge and skills in structuring and programming style. ^[12-16].

Since, most of the students are not able to see what is happening internally when executing a program, in addition to this, a good teaching aid is required to help them in visualizing the execution process and let them to enhance the conceptual understanding of student’s in programming environment ^[12].

7. Objective Measures & Test

The objective measures and test which has been conducted through the Facebook social media from majority of the Afghan pupils is seemed to be more effective and are sorted out as:

- a. No enough background in computer science courses in school, caused to not learn the programming languages correctly
- b. Not enough practical work, no more time allocation for practicing
- c. Lecturer should teach the programming course with one or two practical examples and assign some basic projects to a group of pupils as a team group or pair group
- d. Most of the time problem is in both the lecturer and pupils, if lecturer is professional and hardworking, this can cause the pupils to learn more efficiently
- e. The problem is more in the system and teaching methodology, the slide systems should be ban and books should be the main reference materials, pupils should be loaded with more assignments and practical projects and main point more restrictions needs to be applied on pupils during exam.
- f. Lecturer plays a key role to guide the pupils and teach them how to learn not how to get.(we don’t want to know how to eat fish we want to know how to get fish from the sea)
- g. Having goal and interest in the course is the priority for a student second problem is the huge number of pupils in classroom and unprofessional teaching by some lecturers
- h. As far as I know not paying attention of pupils to the course, however, if the lecturer is not professional still the pupils are able to solve their problem via some online resources such as YouTube, Udemy, and Kaggly and so on.
- i. Lecturer is just acting as a guide this is the pupils responsibility to try more and learn more
- j. Both play an important role (lecturer and student) sometimes the lecture tries more to impart the knowledge but pupils neglect and vice versa, but mostly related to pupils efforts and endeavor if a lecture teaches 20% of the course on the other side the pupils require to try 80% of the remaining course

- k. In my opinion the non-availability of the facilities such as internet, equipped lab and useful curriculum
- l. The main problem is with the pupils a lecturer cannot teach from A-Z this is the pupils responsibility to reach to their goals and learn the concepts
- m. Both can cause the pupils to not learn the programming languages correctly first useless teaching methodologies second pupils less efforts
- n. Programming languages are very tough if It is not learned step by step

8. Conclusion

Achieving the quality of teaching in introductory programming language course plays essential role, hence this course, besides its objectives, prepares the student for further courses, programming skills, problem solving skills, and orient the student attitude to computer disciplines. Unfortunately teaching this course faces a lot of challenges.

It is also been discussed about the readability and Writability of the course where the most significant part to judge a languages is the usability and ease where programs can be understood and read on the other hand Writability means how simply we can create our programs and write our source code for a specific problem, surely stated these two terms can influence reliability.

In addition to this, the challenges faced by pupils while learning programming and the solutions have been spotted, hence, student now will be able to overcome on the challenges they were faced before can easily shows interest in learning any programming languages.

In the last section of this review paper pointed about the problems while teaching this course to undergraduate pupils and the solutions and suggestions which have been given by the pupils of computer science from different universities in Afghanistan.

9. References

1. Anthony R, Janet R, Nathan R. Learning and Teaching Programming: A Review and Discussion. Computer Science Education. 2003; 13(2):137-139.
2. Phit-huan T, Choo-ye T Siew-woei L. Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception. International Conference on Computer Technology and Development. 2009; 9:3892-1.
3. Brian WK, Dennis MR. the C Programming Language. (2nd ed.). India: Pearson India Education Services, 2018.
4. Pauch R, Kelleher C. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice, ACM Computing Survey 37, 2, 2005.
5. Nikolina B, Ivica B. Teaching of Novice Programmers Strategies, Programming Languages and Predictors. Conference Paper, 2015, 1-3.
6. Robert WS. Concepts of Programming Languages. (4th ed.). India: Thomson Press India, 1999.
7. Mohammad H, Bouchaib F. Application Domain and Programming Language Readability Yardstick. 7th International Conference on Computer Science and Information Technology (CSIT), 2016, 1-2.
8. Ravi S. Programming Languages Concepts & Constructs. (2nded.). India: Addison-Wesley, 2000.

9. Peter van *et al.* The Role of Language Paradigms in Teaching Programming. Computers and Education Conference. 2003; 1(3):269-270.
10. Nikolaos SP, Stathis Z. Teaching Programming through Problem Solving: The Role of the Programming Language. Federated Conference on Computer Science and Information Systems, 2013, 1545-1546.
11. Mohamed Osman H, Mohammad A. The Challenges and the Opportunities of Teaching the Introductory Computer Programming Course: Case Study. Fifth International Conference on e-Learning. 2015; 87(76):324-329.
12. Siti Rosminah D, Ahmad Zamzuri M. Difficulties in learning programming: Views of pupils. International Conference on Software Engineering Education. 2012; 76(98):1-4.
13. McCracken M *et al*, A multi-national Multiinstitutional study of assessment of programming skills of first-year CS pupils, ACM SIGCSE Bulletin, 2001; 33(4):125-180.
14. Lister R *et al*, A multi-national study of reading and tracing skills in novice programmers, in ITi CSE-WGR '04 Working group reports from ITi CSE on innovation and technology in Computer Science education, 2004, 119-150.
15. Tenenberg J, Fincher S, Blaha K, Bouvier D, Chen T, Chinn D *et al.* Vandegrift, Pupils Designing Software: a Multi-National, Multi-Institutional Study, Informatics in Education. 2005; 4(1):143-162.
16. Rudder A, Bernard M. Shareeda Mohammed, Teaching programming using visualization, in WBED'07 Proceedings of the sixth conference on IASTED International Conference Web-Based Education. 2007; 2:487-492.
17. Niyazi A, Nuraiym M. Programming Languages. IEEE Software Engineering Education, (ISSN: 4106-3), 2014, 1-3.