

Teaching and Learning System for Programming -A Literature Study to Develop Logic for Programming through Puzzles

Swati Shirsath

*Department of Computer Engineering
Cummins College Of Engineering, Pune, Maharashtra, India*

Abstract- In this paper I describe different teaching methods for programming. These methodologies will include solving different puzzles to develop logic instead of teaching programming technically. It is an approach to make teaching and learning process simple and clear at basic and higher level of programming. By solving different logical games or puzzles while teaching theoretical concepts of programming will help to develop student's logic building capacity, which ultimately will help them to implement complex program statements.

Keywords—Teaching programming; logic development; solving puzzles; logical games.

I. INTRODUCTION

Programming is acknowledged by tertiary educators to be a complex and difficult intellectual activity, with students struggling through their first programming subject and educators [3]. Teaching methodologies for programming at Undergraduate level of Computer Science includes algorithms, pseudo codes, modular or functional programming, divide and conquer methods, etc. Even if above mentioned methods proved to be standard methods for teaching programming, they are not much promising, as an average student is unable to design a code at basic level. John Backus explained functional programming as a tool for teaching programming [1]. But it works well when someone is good in basic programming. Like this there are so many methodologies in the education field. For example, different strategies like divide and conquer, Game Maker Engine [2] which can be implemented but are they promising for a novice programmer? A novice programmer expects simple and interactive methodologies. This paper focus on teaching programming in interactive way that is nothing but playing games or puzzles and then linking technical knowledge with games or puzzles. Well indirectly, how to think about given program statement and how to apply logic for the same using games and puzzles.

II. DIFFICULTIES IN TEACHING AND LEARNING PROGRAMMING

It has been said that a novice programmer takes 10 years to become expert programmer [4]. As students at Undergraduate level comes from different background [6] some students are very visual, some are more auditory or kinaesthetic [7]. Learning style is a preferred learning mode in which students respond to and use stimuli in the context of learning [8]. A staff always starts teaching programming from syntax, algorithm, flow charts, etc. students are totally unfamiliar with all these technical concepts (or words), and so learning and teaching becomes difficult for teacher and learner from stage only. Hence when a small program is statement is given to a student they fail to write code [4]. So algorithm, syntax, data types, variables, etc becomes fixed bugs for their brain system. So learning programming in technical language become an intricacy for a student (novice programmer) in initial phase only and hence the higher level concepts are not easily digestible for them and they fail to apply those concepts while designing an application [4]. Somehow this leads to spoon feeding as everyone is concerned about completing syllabus than learning and teaching programming. Teaching plays an important role at Undergraduate level and should be conveyed in proper way. Mostly C language is used for education though there has been much debate on appropriateness of different languages [9] [10] [11] [12] [13]. I would say teaching and learning programming is a cold war between an expert programmer and a novice programmer and experts are trying to resolve it since 19's using different techniques and methodologies.

Teacher is totally unfamiliar about the knowledge of every student in the class, so he/she tries to teach programming knowledge where strategies to solve a program are expected. Like this there are so many constraints (which are inabilities of students) studied and reviewed by Winslow in 1996[4]. It somehow becomes a difficult task for a teacher to explain basic knowledge of programming to a young novice programmer. Teacher fails to deliver the logic in initial phases and hence students always need support or help while writing a code for a given statement at Undergraduate level.

III. INABILITIES OF STANDARD TEACHING METHODS

A. *Using Algorithms and Flow Charts –*

One of the teaching methodologies for programming is writing an algorithm for simple program statements. A novice programmer uses an algorithm is like someone is getting results before they appear for an exam. Learners should know the logic first and then an algorithm can be written for that logic. Students always gets stuck when an educator ask them to write algorithm for a simple statement. Also they are baffled about the declaration and initialization of variables, inputs, outputs and the flow of a program. The flow of a program is must while writing an algorithm. And for that logic should be known so that it can be fixed into the flow of a program correctly.

B. *Modulated Programming or Functional Programming –*

Module is the collection of data and variable or one can say it is the implementation of program statements in parts. Basically module is difficult concept for a novice programmer. They can't figure out the modules for a given statement as they don't know how to deal with the program statement.

C. *Using Different Strategies –*

Backtracking, Divide and conquer, Greedy algorithm, etc are different strategies which can be used to design a complex program statement. These strategies are implemented for complex program statements using different data structures. But these strategies fail when we talk about teaching programming to a novice programmer because he/she don't have good knowledge of data structures nor they know well how to implement and use data structures for different applications.

D. *Using Interfaces or Templates –*

There are so many interfaces (and templates) available in market for programming. For example, eclipse is an IDE for java, C++, etc. Mostly those are used to design an application in high level language. Those interfaces provide us the syntax and what programmer has to write is a code or logic. Well these interfaces can be used when someone has used basic syntax several times and to avoid consuming time we can go for templates or interfaces. It can't be a good start for a novice programmer.

E. *Puzzle based programming –*

Puzzle-based learning (PZBL) [14] [15] [16] aims to teach students critical thinking and problem solving techniques. This techniques uses divide and conquer strategy where student supposed to divide the problem statement into small pieces and after writing line of code for each piece they have to arrange them in a correct order. A problem should present a challenge to problem solver, having obvious solution, but still showing the promise of resolution [16].

F. *Pair programming –*

In pair programming means two programmers work on same statement. It is found that code developed using pair programming method, is usually completed in less time then when individual students code it, is better designed and has fewer errors. Despite all its benefits, pair programming can sometimes be irritating and exhausting [17]. Because there is always difference between thinking and skills of two programmers and hence affects their team work.

IV. LOGIC DEVELOPMENT

A.. *Need of logic development –*

The three primary pedagogical goals in teaching a programming language are therefore covering the language's syntax, developing program design skills and creative thinking [18]. According to one classification of learning styles [16], learning is divided into deep and surface approach. Deep learning refers to gaining understanding of a topic, while surface learning concentrates on memorizing the facts. In programming, surface learning can be used for memorizing the language's syntax, but deep learning is crucial, in addition to surface learning, for gaining a true understanding of programming logic and consequently a true competence in programming. One can teach programming using different innovative ways but the learner should learn how to write a code using his/her own logic. Syntax and theory of any programming language can be learned or taught easily, the issue is to use them in a proper way. For example, sometimes we have choice to use for loop, do-while, or while loop but if logic says that the iteration should run once irrespective of the condition applied to it then in that case one has to use do-while loop.

B. *Importance of games and puzzles for Logic development –*

Logic development is important as it is the base of all the strategies to program. Logic development can be improved or taught by playing logical games (puzzles). Puzzles like Sudoku, etc and games where one has number of inputs and he/she has to manage those inputs to reach the goal. These games or puzzles shapes the creativity and thinking of a student in a very different way which is helpful for programming. Well these games also helps to decide when and where to use which input. Well starting from very basic logical games (puzzles) one can go for difficult puzzles. So they play an important role in logic development.

V. CO-RELATING PROGRAMMING CONCEPTS AND LOGICAL GAMES (PUZZLES) WHILE TEACHING

Logical games and puzzles should be of very basic level in the initial phase irrespective of level of each student. Once students are familiar with particular game or puzzle, the level of game or puzzle should be increased to next level so that students will come to know how solutions can be modified if little variations are added to previous level of game or puzzle. At least 15 logical games (or puzzles) should be solved by students in the class. Well every programmer (I would say expert programmer or the one who knows to program) knows that logical games (or puzzles) are helpful to develop logic building capability, but we never use them in academics especially as a teaching method. Consider the example; Sudoku is a puzzle which can teach various things related to programming and logic development. Sudoku has some rules which we have to follow like we follow syntax in programming. Arranging 1-9 or 0-9 numbers in a row in Sudoku to get particular output teach us to use inputs in different ways to get a particular output. So the strategy is 1st row should get completed first and then 2nd row and so on. So that solution will be found out in less time. Hence the flow of a program can be taught indirectly to the students keeping in mind that time complexity should be minimum. Well solving puzzles like Sudoku will surely help to build logic development capacity of a student. Now when the theory is started in the class, students can easily correlate both things and also they won't get bored nor will they feel that they are incapable of learning programming. Moreover students can feel as if they are playing games or solving some puzzle in different styles so the percentage of bunking lectures will also reduce.

VI. TEACHING PROGRAMMING CONCEPTS USING GAMING STRATEGY

Consider Fig. 1 where for loop is used to print numbers from 1-100 numbers using C language. There are so many ways to print numbers from 1-100 using for loop only. Well the name of puzzle is for loop. The very first task is to print 1st 100 numbers. First explain the meaning of each and every statement used in that small code (like rules of game) like use of printf, meaning of %d, etc. So that students can display first 100 numbers using simple for loop. Now we can move to next level which is slight difficult to previous one. Students are supposed to print even or odd numbers, so that they will come to know how different conditions can be placed in for loop. And also they can learn to use syntax in different ways without getting any syntactical error. Even when that loop terminates or the flow of code can be easily taught to them rather they will come to know about it more precisely. This will make them more comfortable with for loop. Same procedure can be applied to while loop and do-while loop. But students will have three choices this time. But the difference between all these loops can be explained by designing different small codes. Even data structures can be taught using this strategy. So in this way we can teach all programming concepts to students in an interactive way so that it will not become burden for them.

```
for (i=1; i<=100;i++)  
{  
    printf ("%d", i);  
}
```

Figure 1. Syntax of Basic For loop in C language

Similarly, Data Structures can also be taught through games like Stacks can be explained well using different games on stacks .



Figure 2 . Stack Of Plates

From Figure 2 stack we can easily state that stack operations can be done in FIFO order otherwise the structure will get disturbed. So the program is always written in such a manner so that elements are inserted from 0th position.

VII. CONCLUSION

So here by I conclude that programming can be taught with the help of games ,puzzles etc. And can easily learn with interest which in turn provoke to code a program with minimum no of lines and better usage of system cores. Also it will be the justice with technology as it will be used in smart ways because of good logic development.

REFERENCES

- [1] John Backus, "Can programming be liberated from Von Neumann Style? A functional style and its algebra of programs," IBM Research laboratory, San Jose, vol 21, number 8, August 1978.
- [2] Cristiane Camilo Hernandez, Luciano Silva, Rafael Alencar Segura, Juliano Schimiguel, Manuel Fernandez Paradela Ledon, "Teaching Programming Principles through a Game Engine," Clei electronic journal, volume 13, number 2, paper 3, August 2010.
- [3] Lahtinen E, Ala-Mutka K, & Jarvinen H, " A Study of the Difficulties of Novice Programmers," Paper presented at the 10th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE '05.
- [4] Soloway, E. & Spohrer, J., " Studying the Novice Programmer," Lawrence Erlbaum Associates, Hillsdale, New Jersey. 497 p, 1989.
- [5] Kolling, M. & Rosenberg, J., "Blue - A Language for Teaching Object-Oriented Programming," Proc. of the 27th SIGCSE Technical Symposium on Computer Science Education, pp. 190-194, 1996.
- [6] Wu Y, " Applying a hybrid problem-based learning method to the teaching of computer programming," The China Papers, Issue 6, pp. 63-66, 2006.(references).
- [7] Falkner N, Sooriamurthi R, Michalewicz Z, "Puzzle-Based Learning for Engineering and Computer Science," Journal Computer, Vol. 43, Issue 4, USA, pp. 20-28, 2010.(references).
- [8] Jenkins T, "Teaching Programming - A Journey from Teacher to Motivator," Proceedings of 2nd Annual LTSN-ICS Conference, pp. 65-71, 2001.(references).
- [9] R. P. Mody, "C in education and software engineering," SIGCSE Bulletin, 23(3):45-56, 1991.
- [10] S. Hadjerrouit, "Java as first programming language a critical evaluation," SIGCSE Bulletin, 30(2):43-47,1998.
- [11] R. Biddle, E. Tempero, "Java pitfalls for beginners," SIGCSE Bulletin, 30(2):48-52, 1998.1

- [12] D. Clark, C. MacNish, G. F. Royle, "Java as a teaching language opportunities, pitfalls and solutions," In Proceedings of the 3rd Australasian Conference on Computer Science Education, pages 173–179, 1998.
- [13] R. Close, D. Kopec, and J. Aman, "CS1: perspectives on programming languages and the breadth-first approach.," In Proceedings of the 5th annual CCSC Northeastern Conference on Computing in Small Colleges, pages 228–234. Consortium for Computing Sciences in Colleges, 2000.
- [14] Merrick K E, " An Empirical Evaluation of Puzzle-Based Learning as an Interest Approach for Teaching Introductory Computer Science," IEEE Transactions on Education, Vol. 53, No.4, pp. 677-680,2010.
- [15] Yoneyama Y, Matsushita K, Mackin K J, Ohshiro M, Yamasaki K, Nunohiro E, " Puzzle Based Programming Learning Support System with Learning History Management," Proceedings of the 16th International conference on Computers in Education, pp. 623-627, 2008.
- [16] Falkner N, Sooriamurthi R, Michalewicz Z, "Puzzle-Based Learning for Engineering and Computer Science," Journal Computer, Vol. 43, Issue 4, USA, pp. 20-28, 2010.
- [17] Chaparro E A, Yuksel A, Romero P, Bryant S, " Factors Affecting the Perceived Effectiveness of Pair Programming in Higher Education," Proceedings, 17th Workshop of the Psychology of Programming Interest Group, pp. 5-18, 2005.
- [18] Zacharis N Z., "Measuring the Effects of Virtual Pair Programming in an Introductory Programming Java Course," IEEE Transactions on Education, Vol. 54, No. 1, pp. 168-170, 2011.