*VFAST Transactions on Education and Social Sciences*
*http://vfast.org/journals/index.php/VTSE@ 2018 ISSN(e): 2309-3951;ISSN(p): 2411-0221*
*Volume 6, Number 1, January-December, 2018*

*pp. 27-35*

# IMPROVING PROGRAMMING SKILLS: EFFORT ON SCHOOL LEVEL

MISBAH KHAN [1], SHOAIB FAROOQ[1]
[1]Department of Computer Science, University of Management and Technology, Lahore, Pakistan
Email: misbahkhan1310@umt.edu.pk

ABSTRACT. *Strong Programming skills are need off time now a days. Nations having strong programming skills are leading the world. Developing country like Pakistan Lack in this practice and there are very few resources in schools to integrate new technologies and methods that are introduced worldwide. Programming skills is a big issue in computer sciences field and have very few number of skilled programmers. This paper includes how introducing programming on school level (grade 1-3) will help students to increase programing skills and reasons behind poor programming skills .Secondly, it purpose a method to integrate programming in schools having few resources. Furthermore, discuss why students have least interest in problem solving and coding skills.*

***Keywords****:* Coding skills, learning difficulties, programming in schools, programming courses.

1. **Introduction.** Knowledge of computer programming plays an important role to develop problem solving skills. Its parallel integration throughout every education levels is considered worthwhile [1]. But the problem arise due to very low number of resources [2] available in Pakistani schools. As a matter of fact Introducing programming know how on initial stages could prove a great help. A number of countries are already using this technique [3] . Learning capability of children is higher than elders' [4]. Introducing programming in school provides more chances of better understanding of basic and important concept of coding. Introducing coding at this stage will develop a way of thinking for students. Students will map their problems on computer languages to solve typical issues and for purpose of reusability.

Students on early stage show more interest in learning on computer than traditional study [5]. This could be taken as advantage to divert student attention to this subject and leading to coding. According to a survey [6] students shows more interest in learning when they are taught on computers.

Weak programming skills is a big issue. This is because of lack of interest in coding or programming. Most of the Students first time interact with programming language when get admission in universities on under graduation level. There is so much new to learn in start, student's minds are scattered and dealing with many new things, at this stage absorbing a brand new idea is difficult task for them. Now students are dealing with number of things, and at the same time basic programming course like Programing Fundamentals are offered which builds basics of programming. Students normally do not pay attention to these subjects and learning. They normally fail or pass that subject with a low grade. Baseline is, the base is very weak for whole degree programme and all following semester have advance subjects. Now with very weak basis or knowledge about programming, student find high level difficulty and with time lose all interest in coding. The solution to this problem is to introduce programming in schools

**2. Research Objectives.** Objectives of this research paper are

a) Introduce programming awareness at school level to develop a better understanding of codding at latter stages.
b) Creating interest of students towards coding, even in absence of computers/ fewer resources.
c) Outline the method to implement proposed approach

**3. Background and Related Work.** Integrating programming on school level to increase understanding is a very important aspect to improve coding skills. There have been some publications about introducing coding in local language [7]. The reason behind these tremendous efforts is to increase student's interest in programming on very initial stages. To make this easy tools like Alice [8] are introduced which provide 3-D

interaction environment as it's easy to deal with visual objects then coding. Furthermore abstraction of difficult CS concepts on initial stages is also very important to increase students understanding rather than making whole idea so difficult that students lose interest before even starting it [9].To create better understanding and increase interest of young students it is best way to present learning among students as a game [10].writing code from scratch is a hectic practice for younger students and leads to lose interest, drag and drop is another effort to deal with this problem [11]. In [12] author proposed a visual logic concept using flowcharts to make easy understanding.

According to rule set of education in initial classes it includes maths, English ,Urdu science etc. if they define compute as core subject at very initial level it can help building a way of thinking and problem solving skill[1] via computer languages at very initial stage. If every student know about computer in earlier classes and then on later classes add few more details in curriculum this will lead to with healthy step towards learn programming [16]. As a matter of fact whatever field they chose, students have basic information about Math similarly if we make computer education a trend all students will have basic information and when they will join special courses in universities, at this stage this is not brand new idea. They are already familiar with it, and now they are able to increase their skills with more provided facilities in advanced environment. Introducing novice to new concept is a trick the selection of best and appropriate method is very important in [17] author emphasise on using python as an initial language to introduce programming because it is near to natural language and provides easy understanding.

**4. Integration of concepts.** Introducing a novice to programming needs deep attention. Especially students of grade 1 to 3 need more attention to make learning gametal. This paper purpose a way to make this happen it has two folds. One to make it easy and interesting for novice students to learn basic programming concept and other one is integrating programming in schools with fewer resources.to teach students basic C will be used. The implementation of this idea needs two physical objects

A) Magnet boards.
B) Pile of alphabets, numbers, operators (+,-,*, /, =), semicolons, strings, functions (printf, scanf, output).
C) Awards for students ( shields, stars, cup )

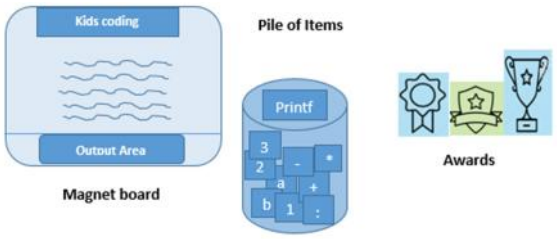


**Figure- 1:** Physical Objects

**Method and process.** Teacher will introduce programming concepts (C Language) theoretically at start of class and after that student will practice that concepts on magnet boards selecting items from the pile.

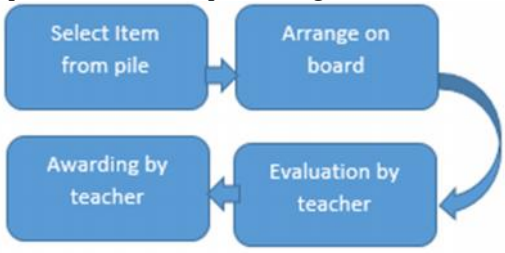


**Figure- 2:** Flow of activities

At the end of session teacher will evaluate each student by checking their boards.to increase interest and make it gametal teacher will award students according to their performance. These awards will be consist of objects like shield, cups and stars. Number of these awarded to student will determine grade of student at the end of year. Student with higher number of cups will get 1st position, student with higher number of shields will get

2nd position and student will higher number of stars will get 3rd position.
The main purpose of this effort is to develop an understanding about programming which will build strong basis. And in further stages like higher classes, universities and then in industry student can contribute in constructive way. For example, flowing is rough sketch of magnet board idea using integer and operators from pile of items selected by students and displayed on board.

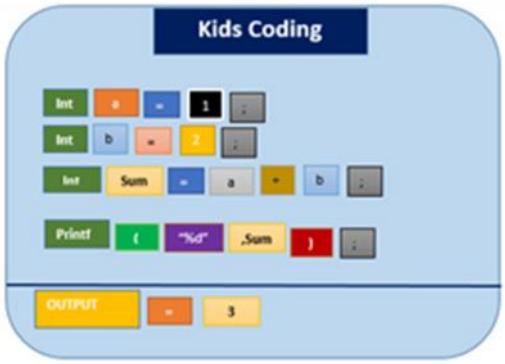


**Figure- 3:** Practice Board

This board could be used for testing purpose. For example teacher can set board like fill in blanks and ask students to fill in or complete the code. This approach will increase student interest and coding will be like a game to them. With this technique programming will be integrated in school without creating labs.

**Course.** This course for grade 1-3 consist of simple stories that helps in better understanding of programming concepts in form of stories books, simple puzzles and drag and drop activities. To achieve this purpose we are using studio.code.org lesson plans.

**Curriculum for Grade 1.** Children on this stage are more attracted to stories and pictorial display of concepts. This is the best way to enhance their understanding about programming concepts. This paper is purposing a story book consisting of programming concepts and book will include following stories (chapters) and also some lab work of very initial level to make students comfortable with computer environment.

Table 1. Grade I syllabus

| **Storybook chapters** | **Lab work** |
|---|---|
| Debugging: Unspotted bug | drag and drop Activities |
| Ali and big project | Programming: Happy Maps |
| Real life algorithms: plant and seed example | Programming in maze |

**Tools & techniques**

Source: https://studio.code.org/s/coursea
Lessons: 1-6

**Instructor skills.** Complete knowledge about each lesson, process and tool. Before starting lesson 15-20 minutes warm-up lesson by delivering verbal lecture.

**Drag and drop.** In drag and drop sections student are ask to drag objects and pace them in right position this help students to be familiar with computers and also help to learn counting as well.

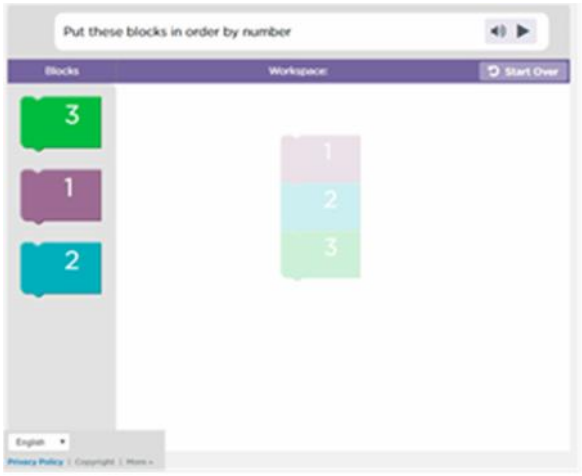


**Figure- 4:** Drag and drop example

**Maze programming example.** In this section student learns to give commands. In this lesson student can also see his/her code which he build by giving commands. Maze lessons are consist of animated character to make it attractive for students.


**Figure- 5:** Maze programming example

**Happy map example.** This is a paper work activity teacher will print out maps for all students and ask them to select the right option. Student can pick right answer by just marking it or on the other hand teacher can ask students to cut arrows and paste them on the map to make it more fun for students.

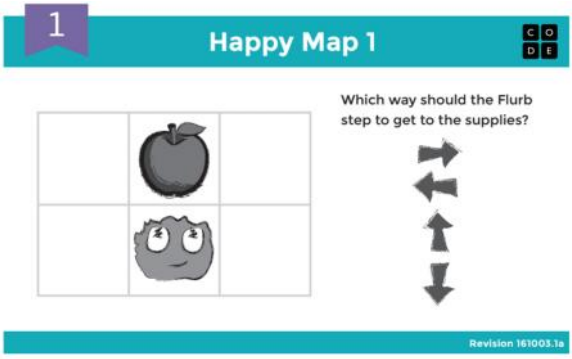


**Figure- 6:** Happy map example

After this course student will get familiar with programming concepts and usage of computers.

**Curriculum for Grade 2.** At this stage students are comfortable around computer and they can learn little bit more complex concepts. In grade 1 student learn about maze and drag and drop in simple words how to give a command to computer how computer respond to commands in this section we will map programming concepts on above idea. Loop and debugging are main parts of this code. Teacher will ask students to do tasks which have many similar steps and then introduce loop to reduce work course will include following concepts.

Table 2. Grade 2 syllabus

| **Lessons** | **Lab work** |
|---|---|
| Loop Introduction | Course A: Lesson 8 |
| Loops in collector | Course A: lesson 9 |
| Loops in artist | Course A: lesson 10 |
| Loops in harvest | Course C: lesson 10 |
| Loopy with ray and BB-8 | Course C: lesson 8 |
| Introduction to nested loop | Course D: lesson 4 |
| Debugging in Maze | Course C: lesson 2 |
| Debugging in collector | Course D: lesson 8 |

**Tools &Techniques**
**Source:** https://studio.code.org/s/coursea
**Instructor skills.** Complete knowledge about loops, how loops works real life examples of loops and benefits of loops. Before starting lesson 15-20 minutes warm-up lesson by delivering verbal lecture. At end 15-20 minutes assessment of activity.
**Loops in collector example.** In loop lessons there are some tasks which require a lot of similar steps to do one task then gradually lead to loop introduction, that how loop reduce work. This is more like game which help young students to learn and play at a same time.

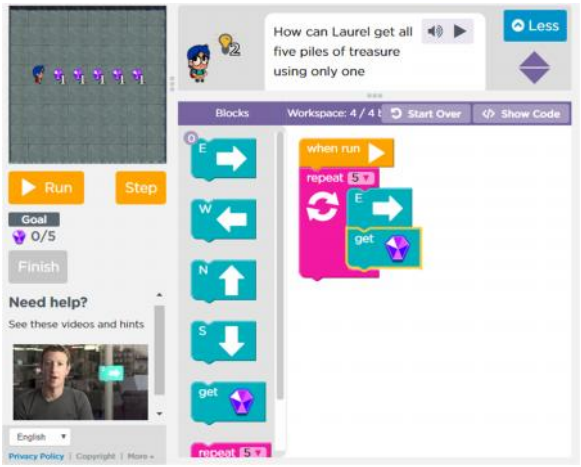


**Figure- 7:** Loop example

**Nested loop example** This section deals with making student's understanding about loop with in a loop. Telling students a complex concept in a very simple and gametal way that they learn while playing.

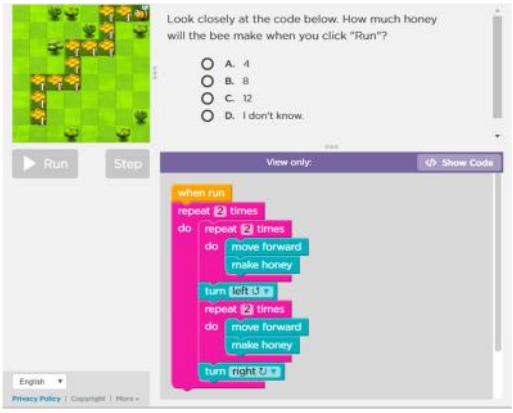


**Figure- 8:** Nested loop example

**Debugging.** This section deals with finding bugs in a given scenario. Some code blocks are already arranged but they do not give required output now student have learn a lot of concepts and practice it several times. At this point they can figure out bugs of given example and rearrange the blocks to get desired output. The process is very simple students can try this endless time but for the sake of competition teacher can also set time for students to find the bug and student who find bugs and rearrange block in less time can win prize. This practice will increase student's interest.


**Figure- 9:** Debugging example

At the end of every lesson there is a quiz to assess the understanding of students. This practice make sure that student are spending productive time in labs.

**Curriculum for Grade 3.** After grade one and grade 2 students are comfortable with tools and environment after repeating each concept several time they have built a strong understanding of proposed concepts. In first two grades they deal with very simple example this grade will include little bit more complex example of nested loops and then introduction to two more concepts including while loop and if else statements. Off course these are complex concepts for young students but according to lessons plans they are presented in very easy and interesting way that student learn at ease. Course includes following lessons.

Table 3. Grade 3 syllabus

| Lessons | Lab Work |
|---|---|
| Revision of nested loops | Course D: lesson 4 |
| Nested loops with frozen example | Course D: lesson 6 |
| Introduction to while loop | Course D: lesson 9 with Real life examples by teacher |
| While loop farmer examples | Course D: lesson 9 |
| Introduction to If/Else conditions | Course D: lesson 10 |
| Conditional in Bee | Course D: lesson 11 |
| Conditional loops with maze examples | Course D: lesson 12 |
| Conditional loops with harvest examples | Course D: lesson 13 |

**Tools & Techniques**
**Source:** https://studio.code.org/s/coursea
**Instructor skills.** Instructor should have grip on concepts of loops and If/Else conditions. He shouldn't only depend on lessons but also giving real life examples to student at the beginning of class. Try to make good understanding of concept before starting on computers once students get what they are going to learn then move them to learn on computers.
**While loop Example.** These lessons make understanding about while with simple examples. Procedure involve same drag and drop technique with pictorial display of action.

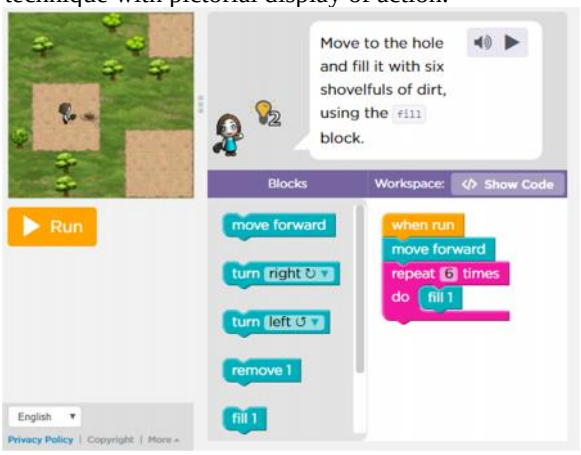


Figure- 10: While loop example

**If/Else Example.** These example a brief introduction to if/else conditions by giving simple scenarios and animated display of actions. This is important to describe these conditions to students verbally by giving them real life example.


Figure- 11: If/else example

**Conditions in loops.** These examples shows how we can manipulate condition (if/else) to loops. These examples help with understanding of combination of concepts to get desired results. This is one step up from basic concepts.

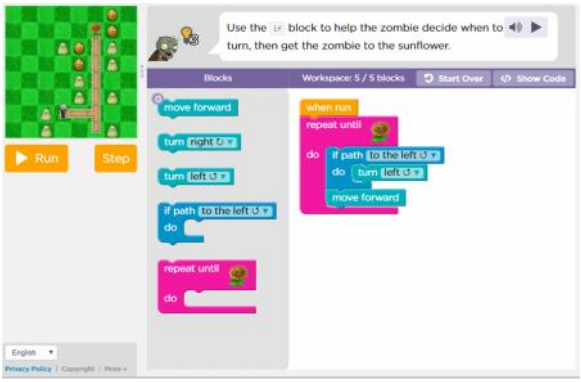


Figure- 12: Conditions in loops

After getting these courses students will learn basic concept of programming then in later grades it will help students to manipulate these concepts in programming using any particular language

**Conclusion and Future work.** This paper gives an introduction to how we can improve programming skills by putting effort on school level. To increase output of CS industry it is compulsory to have high number of professionals in industry. To increase number of these professionals it is compulsory to build strong basis by putting effort on very initial stages like schools. As strong as concepts of students related to programing are, they will be able to give a positive input. This paper describes two different way of learning one by using online courses to build basic concepts using drag and drop technique and pictorial display of every action and the other way is to practice those concepts using specific language.

In future this paper will be enhanced by give details about Courses in each standard, define what degree of information can be included in each course? And Formulation of each course by adding topics. Further we will design story book or grade 1.

## REFERENCES


[1] Donchev, I. (2013). Experience in Teaching C++ 11 within the Undergraduate Informatics Curriculum. *Informatics in Education*, *12*(1), 59-79.

[2] Mahmood, K. (2003). A comparison between needed competencies of academic librarians and LIS curricula in Pakistan. *The electronic library*, *21*(2), 99-109.

[3] Osimo, D. (2016). The 2016 Startup Nation Scoreboard. In *How European Union Countries are Improving Policy Frameworks and Developing Powerful Ecosystems for Entrepreneurs. European Digital Forum*.

[4] Kuhn, D., & Pease, M. (2006). Do children and adults learn differently?. *Journal of Cognition and Development*, *7*(3), 279-293.

[5] Kuhn, D., & Pease, M. (2006). Do children and adults learn differently?. *Journal of Cognition and Development*, *7*(3), 279-293.

[6] Kulik, J. A., Bangert, R. L., & Williams, G. W. (1983). Effects of computer-based teaching on secondary school students. *Journal of Educational psychology*, *75*(1), 19.

[7] Abid, K., Abid, A., Farooq, M. S., Farooq, U., & Abbas, A. (2016). A Strategy for the Promotion of Computer Programming Using Urdu Language in Pakistan. *South Asian Studies (1026-678X)*, *31*(1).

[8] Cooper, S., Dann, W., & Pausch, R. (2000, April). Alice: a 3-D tool for introductory programming concepts. In *Journal of Computing Sciences in Colleges* (Vol. 15, No. 5, pp. 107-116). Consortium for Computing Sciences in Colleges.

[9] DePasquale, P. (2002). Subsetting language elements in novice programming environments. In *Proceedings of the RESOLVE Workshop* (pp. 108-111).

[10] Korpi, J. (2014). Teaching programming to children through games.

[11] Federici, S. (2011, October). A minimal, extensible, drag-and-drop implementation of the C programming language. In *Proceedings of the 2011 conference on Information technology education* (pp. 191-196). ACM.

[12] Gudmundsen, D., Olivieri, L., & Sarawagi, N. (2011). Using visual logic©: three different approaches in different courses-general education, CS0, and CS1. *Journal of Computing Sciences in Colleges*, *26*(6), 23-29.
[13] Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87-97.
[14] Grout, V., & Houlden, N. (2014). Taking computer science and programming into schools: The Glyndŵr/BCS Turing project. *Procedia-Social and Behavioural Sciences*, *141*, 680-685.
[15] Reichert, R., Nievergelt, J., & Hartmann, W. (2001). Programming in schools–why, and how. *Enseigner l'informatique, Georg Editeur Verlag*, 143-152.
[16] Van Merriënboer, J. J. (1990). Strategies for programming instruction in high school: Program completion vs. program generation. *Journal of educational computing research*, *6*(3), 265-285.
[17] Grandell, L., Peltomäki, M., Back, R. J., & Salakoski, T. (2006, January). Why complicate things?: introducing programming in high school using Python. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*(pp. 71-80). Australian Computer Society, Inc..
[18] Farooq, M. S., Abid, A., Khan, S. A., Naeem, M. A., Farooq, A., Abid, K., & Shafiq, M. (2012). A Qualitative Framework for Introducing Programming Language at High School. *Journal of Quality and Technology Management*, *8*(2).
[19] Abid, A., Farooq, M. S., Farooq, U., Abid, K., & Shafiq, M. (2015). A Strategy for the Design of Introductory Computer Programming Course in High School. *Journal of Elementary Education*, *25*(1), 145-165.
[20] Farooq, M. S., Khan, S. A., Ahmad, F., Islam, S., & Abid, A. (2014). An evaluation framework and comparative analysis of the widely used first programming languages. *PloS one*, *9*(2), e88941.
[21] Farooq, M. S., Khan, S. A., & Abid, A. (2012). A Framework for the Assessment of a First Programming Language. *Journal of Basic and Applied Scientific Research*, *2*(8), 8144-8149.