IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

Build or Improve Programming Logic for Beginner

Sunilkumar Chhotelal Kahar
Student, Department of MCA
Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application, Mumbai

Abstract: *Learning programming is essential for individuals aspiring to become computer programmers. It involves instructing a computer to perform tasks through a set of instructions. For beginners, mastering programming correctly and efficiently is crucial. However, many encounter difficulties in starting their programming journey, choosing the right language, and most importantly, building and improving programming logic. This paper focuses on the challenges faced by beginners in developing programming logic and explores solutions to enhance their logical thinking in solving programming problems. The lack of effective programming logic often leads to decreased confidence among beginners when compared to their peers, who seem to create solutions quickly. Understanding the difficulties faced and providing appropriate solutions can empower beginners to overcome obstacles and grow as proficient programmers.*

Keywords: Logic building in Programming

I. INTRODUCTION

1. Learning programming is a basic requirement to become a computer programmer. Learning programming means learning how to tell a computer to perform a task, which is done by a set of instructions.
2. That is why learning programming correctly and efficiently is very important for beginners.
3. During learning programming, beginners are faced with so many difficulties, like what are the things required to start learning programming, which first language should they choose to start learning programming, why logic is not clicking, how to build logic, and many more.
4. In this, we will see one of the problems faced by many beginners, which is "How to build and improve logic in programming?".
5. There are so many students or beginners who quit their field of programming because they think they are not made for it because their friends are creating programming solutions to problems very quickly, and he or she will not create it or may take so much time that they lose their confidence. And this happens because their friend's thought process in programming logic is good, but he or she is not good in programming logic.
6. In this, we discuss the difficulties beginners face when building the programme logic of a problem and why logic is not clicking when they are solving a programming problem. And what are their solutions?

II. CHALLENGES

- **Choice of programming language:** When beginners are trying to learn programming, they are very confused about what programming language they should choose to start programming. Because in the market there are so many programming languages like Java, Python, JavaScript, PHP, and many more, which programming language should they choose? In a market, there are so many programming languages that follow different programming paradigms like Procedural programming, object-oriented programming, Imperative programming, and many more. So which programming language and which programming paradigm will help them in their journey and help them build programming logic is very important.
- **Less Theoretical Knowledge:** In programming, many students think theoretical knowledge of programming languages is not that important in comparison to practical knowledge. Which leads to problems when they are solving any

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

programming problems, and this takes so much time to solve a simple problem. Beginners are jumping from one chapter to another very fast just to solve practical problems or to complete the whole syllabus, but they don't understand that if they don't get all the concepts of a chapter, then how will they solve those tricky problems that are related to those chapters or the topics of those chapters? And therefore, less theoretical knowledge leads to problems in programming logic building too.

- **Less Patience**: Building logic in programming takes time and patience. But in a world of social media where our screen time changes in just 15–30 seconds, this makes our patience very low. whose impact we can see in our day-to-day lives. This effect harms beginners in their journey of logic building. Beginners who try to solve a programming problem, if they don't get the logic of a problem in 30–60 minutes, then they give up on that problem because they think they spent too much time on it, and when they give up, they lose their confidence too. Therefore, patience is a key to programming logic.
- **Hit and Try:** Beginners think about the logic of a problem, make their programming solution direct on the computer, and execute it. If they get any error or wrong output, they try to change their logic and code directly on the computer, and if they get any error or wrong output again, they make some changes on the programme, and they repeat this again and again. They don't understand that this is not the right way to make a programming solution to a problem. They just hit and try on programmes so that programmes just execute and give them the right output, and if they are lucky, some solutions may execute and give the right output for a problem, but by doing this, they are making a bad habit of programming.
- **Practise:** Beginners have less patience, which is why they are not doing enough practise on programming problems related to one topic. And because they are in harry, they are doing 3–4 programming problems on one topic before moving on to another. And practising 3–4 programming problems will not make their programming logic strong. Even though they are doing 3–4 programming problems a day, they don't do them every day. Which leads to problems in building or improving programming logic.

III. SOLUTIONS

- **Beginner's Programming Language:** There are so many programming languages like Java, Python, JavaScript, etc. But the first C language and the second C++ language are good languages for beginners to start with. because the C language covers basic features of all programming languages like if-else statements, for loops, switch cases, etc. And C++ covers all the concepts of OOPs like Data hiding, Encapsulation, Polymorphism, inheritance, and abstraction. And the most important thing is that the C and C++ languages provide memory management features. Other languages like Java, Python, etc. have an inbuilt garbage collector that does memory management, but if programmers have good knowledge of memory management, they can manage it themselves very efficiently in comparison to the inbuilt garbage collector provided by other languages. C and C++ languages help them in the future to understand how other languages will work internally because most other languages are made in C and C++ languages.
- **Make notes:** We know what problems beginners can face if they have less theoretical knowledge. If beginners have good theoretical knowledge, they can think more efficiently about the solution to a problem. That's why it is a good practise to make notes of everything they read or practise on a problem. Beginners must make notes of everything they read. They must make notes of small things. It's advice to not trust your mind; write it down in a notebook. Whatever topic they read and understand, don't think this thing you will remember. Making notes of theoretical things will help them in revision, and whenever they stick to a programming problem and think they are confused or have forgotten any topics, these notes will help them.
- **Dry Run:** Building Programming logic takes time and patience. But beginners try to save time, and that's why they run the programming solution of a problem directly on the computer, and if they get any errors or wrong output, they keep

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023

changing code and execute it. This is called hit and try, and its solution is dry Run. Writing code (on paper) and tracing code manually and finding the correct output is called a dry run. We need lots of practise and patience to overcome mistakes in tracing code. If someone tries to dry run a programme and after tracing the whole code they find its output but when that programme is executed on a computer it shows error or wrong output that means he/she is weak in dry run and in a programme there may be 3 problems: first, there may be any syntactical mistake; second, he/she may be weak on any concept and in the programme there may be any conceptual mistake; and last, there may be any logical mistake. So that's how, by doing a dry run again, they found out where the problem is, and if there is any syntactical mistake, he or she will correct it; if there is any conceptual mistake, they will go through their notes and make their concept more clear; if there is any logical mistake, they will try to make some changes to logic and do a dry run again. That's how dry runs help beginners improve their programming logic and correct their misconceptions.

- **Patience:** Beginners want to learn programming fast, which is why they don't do a dry run because they don't have patience. But they don't understand that dry runs take time in the beginning, but after so much practise, they will improve their programming concepts as well as their programming logic, so overall, dry runs will improve their programming, and that's how they will save so much time in the future. They think dry running wastes their time, but this is not true. They need to understand the difference between wasting and investing time. Beginners try to solve a programming problem, and if they don't get the logic of the problem in 1 to 2 hours, they give up on that question and see its answer because they don't have patience. This will not help them in logic building; they need to push themselves; they need to try to build the logic of the problem for at least 3–4 days. Yes, this is too much time, but at the beginning, it's important to take time to solve a problem. By taking this much time, they will think of different approaches to solving a problem, and this will build their programming logic. This is why they need to have patience.
- **Practise:** In every field, practise is required to become a master of that field. In programming, lots of practise is required to build or improve programming logic. There is a quote: "Practise makes perfect". This quote is perfect for everyone, especially beginners. Beginners need to practise programming problems each and every day; don't skip even a single day without solving a programming problem. In a day, the minimum number of programming problems solved by beginners must be 4-5, and the maximum is more than 20–25. In the beginning, they will think it is not possible to do 20–25 problems on a single day, but they have to push their limits. If they try to solve 20 problems, at least they will complete 10–15 problems, and this will increase their confidence, and that's how they will improve their programming logic by solving so much practise. Now the question is, where will they find that many programming problems? And the answer is that there are many platforms that provide practise problems on relevant topics and their solutions too, e.g., Hakerrank, GeeksForGeeks, Leetcode, etc. These platforms have features to provide questions depending on the level of the user, like easy, medium, and hard. Beginners must start with an easy level and gradually increase it to a hard level.

IV. CONCLUSION

Building and improving programming logic for beginners is crucial. Starting with languages like C and C++ can lay a strong foundation, and theoretical knowledge is vital for efficient problem-solving. Dry runs help identify errors, while patience and regular practice enhance programming logic. By addressing these challenges, beginners can confidently progress on their path to becoming skilled programmers, paving the way for a new generation of capable developers.

REFERENCES

[1]. https://youtube.com/shorts/8Br1VzLbEO8?feature=share
[2]. https://youtu.be/tdQJAyeKVEI
[3]. https://youtube.com/playlist?list=PL7ersPsTyYt1I2qKWkVT8L4521MmEHdeQ

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, July 2023