

## Points for Logic Building

### 1. Solve New Problems Every Day

The first advice is once you solve a specific problem don't repeat it for more than three to four times. Three to four times is ok but then move to the next problem and face new challenges. Let's say you are practicing for printing different kind of patterns, once you practice enough for two or three times, move to the next coding challenge. Keep moving on and try to face a new problem every single day. Your brain has to prepare itself for the new challenge to reduce the response time and get the logic in programming. Solving a problem to print even and odd numbers multiple times for a couple of days won't help you in programming. Learn new things every day, this will also give you exposure to real-life problems and it will help you to write code for new challenging situations or problems.

### 2. Keep Moving On Level By Level

When you start doing programming start with the easy problem from some resources and then move to the next level. Practice enough for a variety of questions at the easy level, then move to some complex program (intermediate level questions) and try to solve a variety of questions for this level. Again move to the next level or more complex problem (hard level questions) and solve a lot of problems. A lot of websites are there like [GeeksforGeeks](#), [HackerRank](#), [Codewars](#), [CodinGame](#) to practice and improve the logics in programming level by level.

### 3. Divide Problems in Smaller Chunks

When you are given a problem firstly try to understand the complete problem and find out what exactly needs to be done. Think about the problem carefully and write down on paper what steps you need to take in order to solve a specific problem. Think about all the case scenarios, steps and according to that write down the input or variables that you need to take in order to solve the problem. For example, you need to write a program to perform the addition of two numbers. Now break down this problem into smaller chunks...

**Step 1:** By reading it you got to know you need 2 numbers and both need to be stored somewhere (in memory).

**Step 2:** To add those two numbers you need operand (“+”).

**Step 3:** To store the addition result you need some memory.

**Step 4:** You need to display the result to the user or you need to use it somewhere else in the program.

Writing smaller steps will help you to map complex programs into smaller manageable chunks. These smaller chunks can be solved individually and then it can be merged to get the final output or actual solution.

## **4. Check Other People's Code**

One of the best things to get better at programming logic is...keep checking the code that others have written. Check code written by other developers on Stackoverflow (largest community for developers) GitHub, Bitbucket or other open-source libraries. Check some great projects on GitHub and learn from it. Check how people are writing the codes and how people are solving some programming problems. When you look at other people's code and use some method or piece of code from there adjusting in your own code to get the solution, you eventually need to think over it and use your brain or logic to solve a problem and get the correct solution. Checking other people's code also helps you to find out the easier solution or various methods for the same problem.

## **5. Make Projects**

One of the most important thing that a beginner or experienced person should follow to get better at programming logic is to make projects. Working on some real-life projects gives you more exposure and experience to become better at programming. You can choose any kind of project to build such as a web app, android app or iOS app. Make calculators, eCommerce projects, personal portfolio or anything that you love to build. You can make any small application or if you are experienced you can build some complex or big project. You learn how the workflow of building a project goes on.

When you work on some project you need to solve a big problem by breaking it into smaller steps. You need to think over it carefully and solve these smaller chunks in order to build the complete project. You use the programming syntax and logic to write down some piece of code in order to solve these smaller chunks, you also move some code here and there to get the correct result, you merge some piece of code, you use the implementation of one function into another one and you face a lot of challenging situations. When you build a project, you go through a lot of difficulties and you debug a lot of problems that help in building the logic in programming.

### **Tips**

**1.** Don't skip the question while solving some exercises. Most of the beginners make a common mistake that they skip some questions and move to the next chapter. Suppose if there are 10 questions and you solved 7 questions (rest 3 questions you skipped because you think it's easy), out of that you solved 4 questions on your own and you checked the answers for 3 questions from somewhere else. Now when you move to the 2nd chapter and start solving the exercises from this chapter you face difficulty because your brain is not trained to handle the variety of questions for the 2nd chapter (this is all because you skipped some variety of question from the previous chapter that was helpful in training your mind to handle the question from 2nd chapter). Do as many questions as you can and train your mind to improve the logic in programming.

**2.** When someone is teaching programming, don't just understand the concept and think that you don't need to solve the problems on your own if you understood

everything whatever the next person is teaching you. You really need to get your hands dirty in code. You can't get better at building logics in programming until or unless you don't practice on your own.

**3.** Do not check the solution immediately. Check the solution when you spent enough amount of time and still unable to solve the problem. Try to solve the problems on your own first and have patience. Once you solved it, check the solution written by other developers.

**4.** When you make some program and you feel that you need to read some theoretical concept in order to solve the problem then please go through the theoretical concept first. The theoretical concept builds the basic foundation and helps in solving the problem.

**5.** *Be consistent.* Don't leave a gap, it's really important. Practice for the programming questions every day. Practicing for three days and leaving it for two days breaks the rhythm and doesn't help in getting better at programming (especially for beginners).