Omkar Somnath Ghotekar Follow
Feb 9, 2021 · 8 min read · Listen

100 Days Of Code - Complete Guide W/ Resources(Beginner)

Want to become a successful software engineer/developer in 2021 and you are dreaming about getting into big tech giant companies?

Or are you someone who is already a working professional but looking for a big move in your career?

So if you are doing preparation then you must have tried to cover various topics. You plan for months, try to follow it, but anyway lag behind in doing so. There are various resources available to do the preparation, but you soon realize that it is frustrating and also overwhelming.

Solution✓

So here I have prepared a practical roadmap **“100 days of Code”** based on a lot of resources and by watching a lot of videos and various articles.

This roadmap will give you a complete guideline to built a strong coding habit and to achieve your goal as a Software Developer. All the things that you need are a laptop, a working internet connections, determination to solve complex problems and a few months of time. These things will help you grow a habit of coding for a lifetime as thus you can become confident for cracking the interviews of big tech companies.

What's 100 days of Code🤔❓

The challenge is a limited time frame based on searching techniques to cover various topics related to crack the big tech companies. It almost takes around 14 weeks or 100 days of time. The topics are divided into 100 days, you just have to follow them and stay committed to coding for 100 consistent days.

I am hereby covering 5–6 crucial topics in 100 days that will let you target big tech companies such as Facebook, Amazon, Flipkart, Google, etc. The list of the topics is given below:

➠ Data Structure and Algorithms
➠ System Design
➠ Operating System
➠ DBMS
➠Object Oriented Design
➠ Cultural Fit

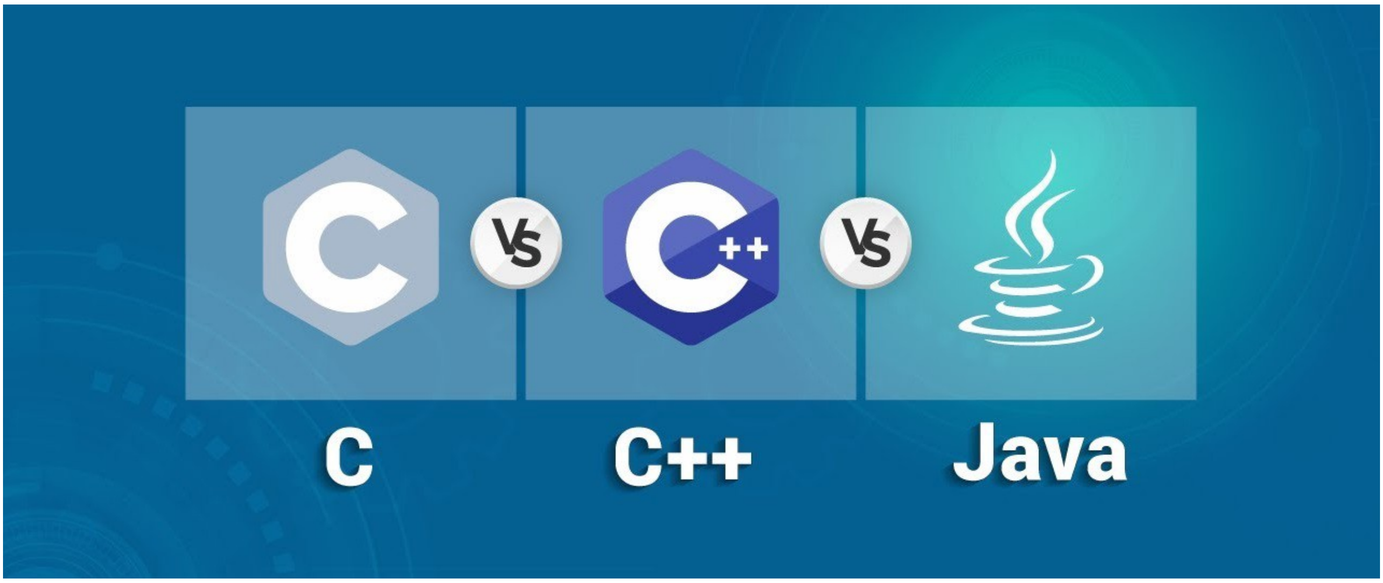
The RoadMap🗺️

Not all things go as you plan, you may face difficulty but you have to understand that, “Great things don't come easily”. These 100 of code also do not include projects and that have to be done separately, so as to show in your resume.

- If you’re a beginner then make the target to solve almost **450–500 coding questions.**
- If you’re an experienced person then make the target to solve a**lmost 300 to 350 questions.**
- System design round is mostly asked from experienced people so beginners have to focus on this section less but experienced one has to focus on this section more.

- Ideally, you shouldn't take more than 20–30 minutes to solve any given problem but keep in mind that this rule is not applicable to all the questions.

Day 1 to Day 10📋



1. Try to solve on your won first even if it takes couple of more hours.
2. Choose a programming language in which you feel most comfortable. I know its difficult but take some time(4/5 days) for it. Its cool.
3. Stretch yourself for a few problems and then go for a hint, but always be aware of the time you are spending on a specific problem.
4. Brush up the skills. Here are some points to remember:-

‣ Reading/Writing from/to files.
‣ Reading input from the console.
‣ Split strings.
‣ 2D array implementation and how to use it.
‣ Basics things of language such as in C/C++ handling null-terminated strings.
‣ Command line interfaces and manipulations from CSV or text files.

Resources to choose language:

1. C Programming
2. C++ Programming
3. Java Programming
4. Python

Day 11 to Day 20📋

Now is the most important part i.e. the Data Structures and Algorithms which technically form the basics of solving any type of problem. These are the most valuable concepts that help you crack the interviews in big tech companies.

All the topics also include their respective links. You can perform their execution in your own code editor. If you know these topics already, get down directly to solve their medium or hard problems, you will learn fast.

I suggest “**Make your own notes**✍”.

1. Strings
2. Arrays
3. Linked List
4. Stacks
5. Queues
6. Time and Space Complexities
7. Trees
8. Tries
9. Graphs
10. Hashing
11. Heaps
12. Sorting and Searching

Congratulations🥳🎉. You have made this far. After completion of all these DSA concepts you will be almost **60%** prepared for the interview stages.

Day 21 to Day 40(Solving Algo’s)

In these 20 days, you would be facing some of the easy problems. Practicing and then leading to perfection is the only key here.

Do Not skip these basic level questions, otherwise you might face difficulty in solving those. Try to solve at least 200–250 questions within the span of 10 days. With a speed of 8–10 questions per day and a time span of 4–6 hours daily, it will be butter on the cake🍰.

I am hereby giving some of the most commonly asked questions on the topics you all have covered in the DSA. Suggest you to try those:-

Array Problems:
Array Rotation
Find duplicates in an array
Missing number in an array
Merge Two Sorted Arrays
Subarray with given sum

Linked List Problems :
Rotate a linked list
Detect Loop in a linked list
Find a middle element in a linked list
Merge two sorted linked list
Searching in the linked list

Stack and Queue Problems:
Sort values in Stack
Implement Two Stacks using one Array

Tree Problems:
Height of a Binary Tree
Minimum value in Binary Search Tree
kth maximum value in Binary Search Tree

Graph Problems:
Implement Breadth-First Search and Depth First Search

Day 41 to Day 60(Half way Reached)

So if you have completed so far, you have already reached half way through the course and will be able to solve almost medium level questions.
Now you should be able to solve almost **150–170 medium** level questions with the span of 20 days. Also depending upon the speed, you can increase the speed of solving of questions. I have provided some of the sample problems. Give them a try!!!

Sample Array Problems:
Kadane's algorithm
Inversion of array
Maximum of all subarrays of size k
Trapping rainwater

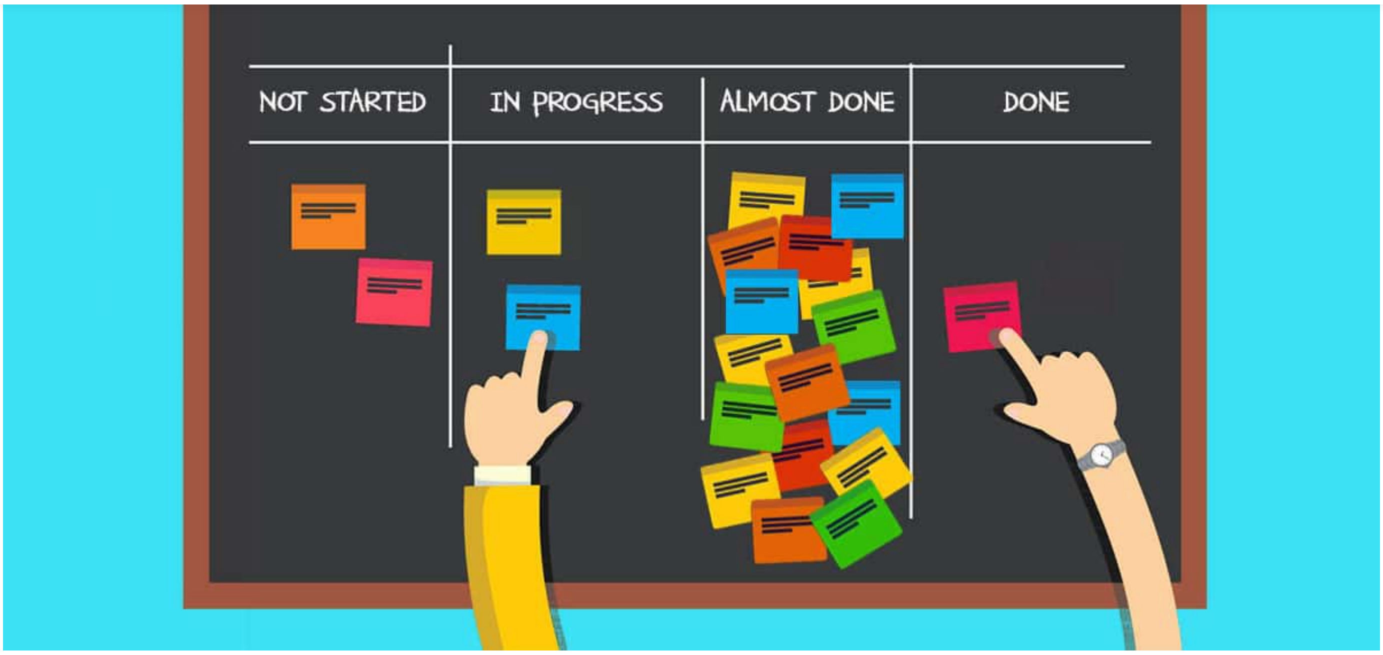
Sample Stack and Queue Problems:
Queue using Two Stacks
Stack using Two Queues
Get minimum element from the stack

Sample Linked List Problems:
First non-repeating character in a stream
Rotate a linked list
Remove loop in a linked list
LRU Cache
Clone a linked list
Find the Intersection point of two linked lists

Sample Tree🌳 and Graph Problems📊:
Detect Cycle in a directed and undirected graph
Find the number of islands
Word Boggle
Implementing Dijkstra
Check for BST
Vertical traversal of binary tree
Boundary traversal of binary tree
Delete a node from BST

Day 61 to Day 75(Almost Completed It)

Once you have got comfortable in solving the medium level problems, then it will be "Icing on Cake🍰" for you to solve the Hard level Problems.
In this further spam of 15 days, I would suggest you to solve almost 70–80 hard problems which comes around **5–6 problems** each



Sample Hard Level Problems:
Sudoku problem
Traveling salesman problem💼
Boolean parenthesization
Find median in a stream
N-Queen problem♟

After you have reached this stage, you will be almost be ready to crack the top companies in their Technical as well as Coding Interviews. But still, some stuff remains which needs to be taken care of.

Day 76 to Day 85(System Design)

This section is usually not given much of an importance, but still to reduce the risk of not getting selected in the interview, just because you haven't read this section should not be there. Hence, go through this sections roughly and understand some basic concepts such as Load Balancing, Caching, Proxies, CAP Theorem, Sharding or Data Partitioning, etc.

Some of the concepts that are most probably asked are:-
1. URL Shortening Service(Tiny URL)
2. Design Uber🚖 or Lyft
3. Design Facebook or Twitter Newsfeed.

This section mostly be descriptive and you can explain these types of terms in your own language to the interviewer. The main objective of the interviewer is that if the candidate is been able to build a large or complex or not.

Day 86 to Day 92(OS and DBMS)

In this part of **#100DaysOfCode** you will learn the most basic and important subjects of Computer Science like Operating Systems, Database Management. Most of the people struggle to study these concepts, but this section is equally important for cracking interviews.

For time being, I have listed out some of the sections most probable likely to be asked in the interview:-

- **Process and Process Management**
- **Thread and Concurrency**
- **Multithreading**
- **Lock🔐**
- **Process Scheduling**
- **Synchronization and Deadlock**
- **CPU Scheduling🖥**
- **ACID Properties**
- **Indexing and Scaling**
- **Distributed file system**
- **Memory management**
- **Critical section problem**

I will surely list down a separate article for the coverage of OS and DBMS further and link it down here. Stay tuned!!!

Day 93 to Day 100(OOD & Cultural Fit)

After coming so far, you will have to prepare yourself for software design principles, SOLID and DRY principles, OOPS concepts, etc.
Some of Object Oriented Design concepts are:-
1. Design a Movie Ticket Booking System🎦
2. Designing Vending Machine
3. Designing an ATM🏧, etc.
The main goal for the object oriented design is to make sure that you truly understand classes, interfaces and then transform them into comprehensible object design. This also tells the interviewer that you clearly understand the class diagrams, activity diagrams or case diagrams.

Companies always hire a candidate which is able to mold himself/herself according to the situation, atmosphere and place, who is always passionate about their product and the culture of the company.

Even if you pass all the stages of interview, but are not culturally fit for the company, there are chances to get rejected. Make sure that you are responsible and have the right attitude.

Hope it helps!!!

Resources and Links:-
GeeksforGeeks
Google

Feel free to connect with me over LinkedIn or Twitter or Instagram for any further queries.

Also check out my YouTube Channel. Soon gonna start something exciting.

Not been able to complete the **#100DaysOfCoding**, get started with **#30DaysOfCode**. Link below.