

23/03/2023  
Thursday

## O. CODING PLAN - JAVA

- 0) Watch the youtube playlist and practice it on a compiler. Basics and Fundamentals of Java.
- 1) Solve 30-40 problems in the given list and try to achieve 5★ in HackerRank. It will strengthen the basics and Fundament and I can move to High-Level-DSA. Also can Showoff my DSA Skills in my Resume/LinkedIn.

After completing this, Solve the given 2 LeetCode Problems.

- 2) Watch the Java Collection By Anuj Bhaiya on Youtube

DSA PRO-PLAN (Make a Revision Doc for each topics - problem, code, approach)

- |                                      |                               |
|--------------------------------------|-------------------------------|
| 1. Hashing                           | 11. Recursion.                |
| 2. Two pointers                      | 12. Binary Trees.             |
| 3. Sliding Window                    | 13. BST                       |
| 4. Linked List                       | 14. Dynamic Programming.      |
| 5. Array Warmup.                     | 15. Graphs and general Trees. |
| 6. Matrix Warmup.                    |                               |
| 7. String, Binary Search             |                               |
| 8. Search and Sort<br>Priority Queue |                               |
| 9. Greedy                            |                               |
| 10. Stack and Queues.                |                               |

Resume (use given template)

LinkedIn

# 1. CODING INTERVIEW

- 1) Don't act like a robot. Little informal connect is needed (not over friendly). Name, college, ... project ... Intern ... Job. Don't say it like reading from somewhere else. Say it with some expressions, emotions. Maintain smile during some moments.
- 2) Don't Stress. Interviewer is also a human as well. It will be helpful while solving problems for the interview.
- 3) Always use good Variable names for solving the given problem.
- 4) Write appropriate functions where needed and write properly structured code.
- 5) Always follow indentation.
- 6) While writing code, explain each and every line/procedure simultaneously. This creates a good impression on the interviewer as they are able to know your communication skills by the way you are explaining as you write.
- 7) Never jump into implementation/coding directly. First explain the solution idea clearly to the interviewer.
- 8) As soon as interviewer gives you the coding problem, ask all your doubts, assumptions and requirements from the interviewer.
- 9) Explain the solution, not just orally, but by writing few things in the space given by the interviewer in Virtual boards given to you.
- 10) Also, use some example input, output while explaining the solution.



## Example:

Problem: Find the number of pairs whose sum is  $X$  in an given array.

Step-0: Don't be silent. Ask questions even if we don't have any. The other person thinks you are a good observer and fit for their team.

Questions: 1) All the integers positive/negative? Range of integers.

2)  $\text{Size}(\text{Array}) \geq 2$ ?

3) What to return if size is 0?

Step-1: Talk about the problem. Start with giving a test case for the problem.

$arr = [1, 3, 4, 5, 6]$      $X = 11$

$\text{pairs} = (5, 6)$ .     $\text{No. of pairs} = 1$

So the answer will be 1. Since there is only one pair.

Then say I have 2 approaches. Brute Force and optimised approach with the space and time complexity. Also while saying this try to write this in the space given in a short and detailed way.

Step-2: coding.

The namings of the Variable should be perfect/suitable for the given problem. Try to write everything in a indented manner and use functions wherever required. This makes the code look good. While writing each line of code speak clearly and explain everything. Use comments within the code if needed. Starting, say: Sir/madam I may need 1-2 mins to process the question. Then I will discuss my doubts and approaches with you. Also tell the programming language we are going to use.

~~int count(int[] numbers) {~~

int count(int[] numbers, int value.) {

Before thinking, ask the interviewer that i need some time to think and write the pseudo code on a paper. So that you can refer that while writing and explaining the code.

Before submitting the code ask the person the time to crosscheck. And go through the code.

If there is Time left. Dry run the code. Dry run is considered as actual testing of the code.

Also ask about CORNER CASE. What should i return if the size is 0 or no pairs found?

At last, the person will ask you to ask some questions. So make sure to always ask the questions.

- 1) How did you join xyz and is ur journey so far.  
How u reached side a.
- 2) What tech stack / domain / team you worked in xyz.

## 2. NON-CODING-SESSION

- \* No. of LinkedIn connections should be high (2000-5000)  
Daily we have to send. Persons who work in a good company
- \* LinkedIn Visibility Principle. Try to showcase yourself in it.
- \* LinkedIn Application. Make sure to apply for job daily and give interviews (5 times a day)



- \* Good Resume. Don't boast about yourself. Let the resume speak.
- \* LinkedIn profile Improvement. Fill up the available fields/sections. All the coding profile and Show offs should be there in the featured section. Also feature the Resume on LinkedIn.
- \* LinkedIn Algorithm
- \* Dev/Job/Interview.
- \* Communication Skills.
  - Clarity and depth of explanation.
  - Psychological Tricks.

Try to speak little louder and confidently

"Hello sir I am Thomas Siby, I have done bsc in Statistics from XYZ. And during 3rd of college, I have explored Java and DSA in depth. In DSA I have explored these these topics in this this platforms. In dev I have made projects these these on tech stacks these these. I have done internship in abc in this this tech stacks. courses certificates. At the last of our Introduction say. Do you want me to explain my projects in detail?"

So by doing this the person will only ask questions related to what you mentioned. And you will be able to explain things. More easily.

What if the interviewer ask about a topic which we don't know? Sir I have not explored this topic. But I have explored these and these very well.

The stacks we have used in our projects. Try to learn and read the interview questions based on these stacks.

Also explain the projects in depth and detail. Also try to find the loopholes/cross questions about your projects.

Also prepare for the question what are the challenges faced by you while working on this project. So add some masala and give answer.

Always talk about the ~~chatter~~ technical things. Not about the non-technical things.

\* Mindset: Don't run for the money. It will affect your work quality. Keep on trying and practising. You will reach your goal.

\* Health and God.

Eat Home Food Always. Drink Water. Do exercise.

\* Resume Trick. Always make a Resume. Different phone number and email id for both. 1 for Resume and 1 for referral. Chances will be high.

\* How to Take Referral? Send referrals to more people.

"I am XYZ and I have done XYZ in XYZ branch from XYZ college. Refr the intro text....."

Resume Link. IF you find me eligible do refer for me this position."



## RESUME GUIDE

- \* Education : Only UG is required and above it as well. Also mention your grade/percentile.
- \* Experience : Write about recent roles, in reverse chronologically (last to first) and the top 3-4 projects which are most relevant to the role. Use STAR format (Situation, Task, Action, Result) for projects, with 2-3 lines per project. In action mention your individual contributions more than what the team did (not what we did, what you individually contributed or led the team to do). In Result, use quantitative metrics wherever applicable (x% accuracy, \$y cost reduction, n% ROI).
- \* Projects : Use metrics. Stocks used and learning by results. Keep only top 3-4 projects relevant to the role.
- \* Skills and courses : Can be from college/online (add only reputed ones). For skill don't add something unless you have hands on experience.
- \* Awards and Achievements : Competitive coding contest ratings, Academic prizes, Hackathon rankings and other prestigious research programs. Write precisely and briefly, "X rank/percentile in Y contest". These should be in our domain.

# Dev - ROADMAP

## FRONT-END:

- 1) Fundamentals: HTML, CSS, JavaScript
- 2) CSS Frameworks: Makes the website responsive:  
Bootstrap, Tailwind CSS, Foundation, Bulma, Skeleton
- 3) CSS Preprocessors: SASS, Stylus
- 4) JavaScript Frameworks: Angular, React, Vue
- 5) State Management Libraries:
- 6) Package Manager: NPM, To install some libraries.
- 7) Version Control System: GIT - To track and manage changes to software code.
- 8) Testing.
- 9) Deployment: Github, AWS cloud, Microsoft Azure
- 10) Advanced Topics: API's.

- Revise HTML, CSS, JavaScript.
- Projects for Practice (Not for Resume).
  - Moving CAR
  - Cheatsheet Template
  - Snake Game
  - Tic Tac Toe
  - Analog Clock
  - 5 project.
- Learn Angular / React.



## BACKEND DEVELOPMENT

NodeJS: create, open, read, write, delete and close files on the server. Can collect form data. Can add, delete, modify data in your database.

DataBase: MySQL, MongoDB.

Java Framework: Spring-Boot, make projects.

### Projects For Resume

Before starting the project. Write the things we discovered. Features. If we have multiple features in mind. Select the best feature and write why u chose this feature. Writing these things down will save time to prepare for the project explanation. Before the interview. Also mention the difficulties and challenges faced.

### Projects in Resume

- Heading (max 4 words). The heading should explain the idea.
- Summary (max 4 points). One point will be the Tech stacks used. Rest 3 points will be the description, short and precise.
- 3 projects should be in our resume.
- Why you are making the project?
- Project should be based on a Real-Life problem.