CDC Preparation Resources

# Part 1

## ML/DL/AI/Data:

1. Revise 'C' & learn C++ / Java & Python. Practice a lot. It will help in the long run. If possible try to learn 'R' but only after Python.

2. Complete 'Analytics Edge' by edX.

3. Complete the 'Machine Learning' course by 'Andrew NG' available on 'Coursera' or Youtube.

4. Complete Five modules of deeplearning.ai ([https://www.deeplearning.ai](https://www.deeplearning.ai)) Check it out, it's really a good one. It's also available on youtube.

5. You can learn Data science at Datacamp at your own pace using the free 3 month individual subscription through the Github Student Developer Pack.

6. Create an account on 'Kaggle' & take courses on Data Analytics there.

7. Create an account on 'Analytics Vidhya'. They keep updating about various competitions happening throughout the year.

# 

## Supply Chain/Finance/Consultancy/UPSC/Others

# 

1. Consulting -

- Case Books by different B Schools like ISB, IIM-A, IIM-C, IIM-L etc.

2. Supply Chain -

- Supply Chain Management by Chopra and Meindl

- Group Discussion Prep Videos on Youtube

3. Product Management :

- Cracking the Campus PM Interview

- Product Management Resources

4. Finance:-

- Level 1 Books of CFA/FRM

- Options, Futures and Other Derivatives by John C. Hull

- Finding Alphas by Tulchinsky

- Aptitude Questions and Puzzles on Gfg

5. Investopedia

6. Soft Skills for Interview Preparation by Communique:

7. Zerodha Varsity

8. Virtual Stock Markets

9. Consulting Fundae

10. KGP Life Fundae

For those working in research groups, do your project well. You people will have a higher chance to make it to Inter IIT tech meet in years to come. Your learnings will help you a lot as an individual & also for the Institute.

For those who are interning somewhere, please work diligently as your rise in the learning curve, knowledge base, & growth should be your utmost priority. Always keep in mind that you represent your institute at the organization you are working with & you can’t let it down.

Apart from this, revise your concepts of Mathematics 1 & Mathematics 2 taught in your first year. It helps a lot. Try to learn a new language. You may use ‘Duolingo’ for the same. It’s a fantastic application. Try to learn a musical instrument if possible. It acts as a stress-buster in difficult times. Play any sport. As you are at home, try to improvise your biological clock. Stay healthy and happy.

---

# Part 2

This Guide is not exhaustive but just a brief overview of how you can start with the preparation. Take your time and decide what field you are planning to target.

If it’s a software profile that you are targeting then start doing questions from interview bit or geeks for geeks. Try doing all the questions from this link by the start of the next academic year.

([https://www.geeksforgeeks.org/must-do-coding-questions-for-companies-like-amazon-microsoft-adobe/](https://www.geeksforgeeks.org/must-do-coding-questions-for-companies-like-amazon-microsoft-adobe/))

Consult seniors who got an internship in a software profile for better preparation methodology. Again for people targeting seniors of that field are the ones who can actually help you.

Now, if you are targeting any Finance/Analyst profile then the following points are a must.

## **1. Probability & Statistics:**

a. Complete all the questions from \*50 challenging problems in probability\*

b. If you want to study the theory through videos then follow these lecture, of course, Stats 110 ([https://projects.iq.harvard.edu/stat110/home](https://projects.iq.harvard.edu/stat110/home))

c. If you are a book person then follow the \*Prob & Stats book by Montgomery\*

d. Dice Problems - [https://www.madandmoonly.com/doctormatt/mathematics/dice1.pdf](https://www.madandmoonly.com/doctormatt/mathematics/dice1.pdf)

## **2. For Puzzles:**

a. \*Heard on the Street\* is a must-do book

b. If you get additional time and want to leave no stone unturned then do questions from these links too.

i. Brainstellar

ii. Geeks for Geeks Puzzles: [Link](https://www.geeksforgeeks.org/puzzles/)

iii. Interview-bit Puzzles: [Link](https://www.interviewbit.com/puzzles/)

Quant: Resources by Quant Cub, IIT Kharagpur

Resources by Code Club IIT Kharagpur:

Must Read for Software Profile

Reading material for Software/Quant Profiles

1. CP: [https://cp-algorithms.com/](https://cp-algorithms.com/)

2. CP Handbook:

[https://drive.google.com/file/d/1Ul\_5yAdnDrtxTcrhwQJi8OjZFRNR8Nr/view?usp=sharing](https://drive.google.com/file/d/1Ul\_5yAdnDrtxTcrhwQJi8OjZFRNR8Nr/view?usp=sharing)

## **3. Cracking the Coding Interview Book**

## **4. DSA: NPTEL Courses**

## **5. Core CS Concepts:**

- For 3rd yrs: Mainly OOPS added to 3rd years. Some specific topics in OOPS that were asked in 2020 are access modifiers, virtual function, overloading vs overriding, encapsulation, abstraction.

- For 4th yrs: 4th years need to mainly focus on OOPS, OS and Networks. A more thorough preparation might include DBMS as well. The portions covered in the academics are more than enough to be decently prepared

## **6. ML:**

Mostly questions from traditional ML, probs and stats are asked. Understanding the classical ML techniques and their intuition is necessary. Domain-specific questions are also asked mainly based on your resume. So make sure you have a good understanding of your intern projects, papers, etc.

## **Topic wise for coding**

1. Binary Search
2. STL
3. Bit Manipulation
4. Maths
5. DP (Dynamic Programming)
6. Graphs
7. DP on trees

## **Experience from a Software Role**

You should be somewhat confident in one or more languages, and you might have developed some intuition for solving the questions. Now, this is probably the time to brush up some of the topics which you fear the most and to revise the rest. But there is a lot more to the interviews. So let's start:

● Revise: Mostly, if you have done a lot of coding, then solutions will come intuitively, but what about some questions with unique answers. These answers could even be exclusive to just save on the space. So for these, go through interviewbit topics once and check if you remember those. If you don't, then don't panic as it is better to not remember now than in the interview itself.

● Data structures overview: Yes, you do remember that you can solve a problem using Set ( in C++ ) but what is the data structure in it? Yes, the interviewer might ask this to understand whether you know how the data structures work. Don't worry about learning how a binary search tree is balanced, but rather just know how it works ( no one will ask you to implement it, hopefully ).

● Pointers: What was the last time you implemented a linked list? Or trie? Or trees with Node\* child? It is common not to know how to apply these, or maybe you forgot or worst case you are afraid of them. But relax, it is easy actually if you don't procrastinate. Give a day or two just to understand how this works and you will have that confidence back.

● CV: It is excellent mentioning all the courses in the CV and the projects that you did. BUT, can you make the interviewer believe its importance and all your efforts there? Some things in CV might be many months older, and you might not remember them correctly at first, so just go through it once again and be prepared for any questions regarding them.

● Communication: This is surprisingly, very important. You don't want to leave a wrong impression on the interviewer where you thought it was your confidence, but the interviewer took it as arrogance. Yes, it happens. Also, what if the interviewer doesn't get what you explained. Tough stuff! So for this, take some mock interviews and get an outsider view about your communication skills.

● Mock interviews: This is the final piece and a must. You will face a lot of pressure in the real interview, and it might result in you forgetting things and panicking. So, for this, you can simulate that experience with the mock interviews. These will also help you to understand your communication skills. And finally, you might get some fantastic ideas to improve your interview experience.

I guess these do cover all the major points needed for the pre-preparation for your interview. Now, coming to some tips for during the interview:

● Relax: Get proper sleep and food before the interview for sure. It will help you all the energy to give your best.

Validating question: Describe the question again in your words to make sure that you are on the same page as the interviewer, and you got the question right. You don't want to spend 10-20 minutes on the wrong question.

Ask for an example: Another step to be sure that you understood the question correctly is to ask for an example to the question.

● Validating inputs: Before jumping to solving the problem, think about the edge cases for the question, and do ask what input can you expect. At times expecting integer output might feel straightforward, but still, ask if it is okay to expect the output to be within the bounds of an integer. Ask whether the input array can be empty or not, and so on.

● Start with brute force: You might get the optimal solution at once, but they came to see your thought process. Start with a general basic solution to show how you improve that. Also, you don’t want to show that you have just crammed up many solutions and this is one of them, so show them your thought process from start instead.

● Keep on optimizing: From your previous non-optimal solution, now start to move towards a solution with a better time and space complexity ( yes, even space complexity matters! ). Don't panic if you feel you are stuck, write some more examples, and try to see some connection between them. Feel free to take 5-10 minutes for that as you got at least 45 minutes.

● Give voice to your thoughts: Interviewer cannot read your mind and understand what trouble you are facing or what fantastic solution you will have. So, be transparent and tell them what you are thinking. It could be stupid too, but won't it be impressive that you finally got an excellent solution at the end while you were stuck at something stupid at the start? This will also help the interviewer to give you some hints if you are going in the wrong direction.

● Hints are not bad: The interviewer gave you a hint, so you failed! Not true at all. You are not expected to be perfect and have the most optimized answer for every question directly. So, take it constructively and try to use it to get the job done. It will not mean that you failed in any way.

● Explain in layman terms: You said that you will use a set as you code in C++, but the interviewer might not know it and might expect something like AVL tree or red-black tree ( No, they won't ask you to implement them ). Different languages might have different implementations, so try also to mention the fundamental data structure behind the inbuilt structures you are using. ( if needed ).

● Coding: You cleared the part to get to the solution, and now finally, you have to code. The interviewer will read the code, and they must understand what you are doing, and so the best coding practices should be followed. Start with writing a skeleton of the function for the question first and ask if it’s okay. Keep the code clean. Reuse the code. Have variable names that directly imply their use. Keep code properly indented.

● Mention the final time and space complexities below the code. ( it looks good )

● Dry run on an example: Mention the interviewer that you will dry run on some example, so to check it works correctly. This helps you explain to the interviewer how your code works. And also you can find some edge cases you missed or some mistakes you made. You can also go a dry run on more examples to check for any edge cases left.

## 

# 