An Exhaustive Guide for Excelling in Academics and CDC Internships and Placements at CSE, IIT Kharagpur

How to Use This Document?

This document has been written by Suhas Jain and Ashutosh Kumar Singh. We have tried to provide all resources we have utilized or found helpful during our 5 years at IIT Kharagpur. Hope it serves its intended purpose by helping juniors and upcoming students in the department.
This doc is both a repository of valuable resources and a roadmap for navigating academics as well as CDC. The sheer amount of content to study might seem overwhelming, but if it is done judiciously and as intended, then a lot can be covered in 4 or 5 years. It is also wise not to try to do too many things at a time. Pick a stream and focus on that. There are multiple resources that overlap or are alternatives to each other. Rather than doing more things, focus on doing justice to whatever you pick.

For First-Year Students

There should be two primary goals in the first year: getting a good CGPA and getting started with programming. It is a time when students should explore a lot, make friends, and get accustomed to the college environment. Not providing resources for first-year academics here, as there are already various freely available channels.

How to Get Started with Programming?

The aim is to learn the basics of C and C++ at least and start to understand how to use your laptop for coding related purposes. Dual booting with Linux (Ubuntu) is highly recommended if you have a Windows laptop.

- Dual Booting with Ubuntu
- C Programming Tutorial
- C++ Programming Tutorial
- Python Programming Tutorial (if interested; after learning C++)
- Please check out the competitive programming section and get started if you have free time.

Courses and Equivalents

The playlists of courses at IIT Kharagpur and their equivalent courses at some of the other top universities in the world have been provided in this section. Watching the lectures of courses taken in universities like Stanford, CMU, UC Berkeley, etc., usually provides a much deeper understanding of the subject, the benefits of which go far beyond just getting a better CGPA.

Note: The courses that might also be relevant for CDC are marked green.

Depth Courses

Course	IIT Kharagpur Lectures	Alternative Lectures
Discrete Structures	Discrete Structures - Autumn 2020	Discrete Structures - MIT Problems
Algorithms - 1	Algo 1 - Autumn 2020 Algo 1 - Spring 2021	Algorithms 1 (1st half) - Stanford Algorithms 1 (2nd half) - Stanford (must watch)
Probability & Statistics	Probability & Statistics - Spring 2021	Stat 110 - Harvard (must watch) Notes Problems
Formal Languages and Automata Theory	FLAT - Spring 2021	-
Software Engineering	Software Engineering - Spring 2021	C++ - Cherno (GOAT C++ playlist; must watch)
Switching Circuits and Logic Design	SCLD - Spring 2021	-
Computer Organisation and Architecture	COA - Autumn 2021 COA Lab - Autumn 2021	COA - ETH Zurich Same prof at CMU (older)
Compilers	Compilers - Autumn 2021	Compilers - IIT Delhi
Algorithms - 2	Algo 2 - Autumn 2021	-

Operating Systems	OS - Spring 2022	OS - UC Berkeley (must watch)
Computer Networks	Networks - Spring 2021 Networks - Prof. AG - Spring 2022	Networks - Northwestern
Database Management Systems	DBMS - Spring 2022	DBMS - CMU DBMS - UC Berkeley
High Performance Computer Architecture	HPCA - Spring 2020-2021	HPCA - Georgia Tech Part 1, Part 2, Part 3, Part 4, Part 5, Part 6

Elective Courses at IIT Kharagpur

The provided list contains the playlists of the courses held during online semesters.

- Machine Learning - Spring 2021
- Linear Algebra for ML & AI - Autumn 2021
- Principles of Programming Languages - Spring 2021
- Advanced Machine Learning - Autumn 2020
- Algorithmic Game Theory - Autumn 2021
- Artificial Intelligence - Autumn 2021
- Artificial Intelligence - Foundations & Applications
- Deep Learning - Foundations and Applications - Spring 2020
- Parameterized Algorithms - Autumn 2020
- Theory and Applications of Blockchain - Autumn 2021
- Information Retrieval - Autumn 2021
- Data Analytics - Spring 2021
- Theory of Computation - Autumn 2021
- Advances in Operating System Design - Autumn 2021
- Usable Security and Privacy - Autumn 2021
- Cloud Computing - Spring 2021
- Computational Geometry - Spring 2022
- Deep Learning - Spring 2022
- Information Retrieval - Spring 2022
- Big Data Processing - Spring 2022
- AI for Economics - Autumn 2021

Elective Equivalents

- Machine Learning - Stanford (must watch if you want to work in anything related to ML)
- Artificial Intelligence - Stanford
- Deep Learning - Stanford
- Natural Language Processing - Stanford
- Cryptography - Christof Paar (very useful if you take the cryptography course)

Bonus

- Missing Semester - MIT (2019, 2020): Contains lectures on certain common tools which are usually not taught in a course
- StatQuest: Goto resource for understanding anything related to probability, statistics, ML, DL, etc., an extremely underrated but amazing resource
- Time Series: This might be useful if working on some quant-related project/intern
- 3B1B - Linear Algebra: The BEST and OG channel for math-related topics
- 3B1B - Calculus
- Large Language Models - Databricks

Still can't find what you're looking for?

Try finding at Open Source Society University - Github: a repo of free online alternatives for CS courses; try this only if you can't find an equivalent course in the above lists.

Resources for CDC Internships and Placements

This section will mainly focus on opportunities in software engineering, systems, and quant domains. Mastering DSA and having competitive programming experience plays the most significant role in landing a good offer in any of these fields. Good CGPA and CP skills open the doors for most day-1 interviews in CDC. This section will also focus on providing resources and a roadmap for doing well in these interviews.

Software Engineering

Competitive programming is the most important thing for getting a software engineering internship/placement. The other relevant thing might be OOPS concepts. This section will focus on how you can get good at it, and all the other stuff that is required. Almost all companies first conduct a coding test and then release their interview shortlist.

Coding tests of software companies almost always consist of just CP problems, and even their interviews are almost entirely focused on CP/algo problems.

Competitive Programming

Get Started

- Learn C++ - C++ is the best for this purpose. Learn it.
- Learn (Practice, practice, practice) STL - Once you have learned C++, you should know STL (Standard Template Library), which provides containers for various data structures and is extremely important. You should know the methods provided by each container, and that only comes through practice.
- After this, you should be able to solve fundamental ad-hoc problems (Codeforces A Ladder, Codeforces B Ladder).

Learn Algorithms and Data Structures

- You can get an idea of the broad list of topics that you should know from Striver's A2Z DSA Course/Sheet.
- Cp-algorithms is also an extremely valuable resource for learning any topic and getting corresponding code snippets.
- At this point, you should simultaneously start participating in contests on Codeforces and AtCoder.

Practice

- Practice, practice, practice - give contests, upsolve problems you could not solve during the contest, and keep practicing problems topic-wise.
- Codeforces - Best site for CP. Keep participating in contests here. It also has exceptionally great blogs for almost all topics.
- AtCoder - Best problem statements, short and simple. You should attempt all AtCoder beginner contests.
- CSES - Solve ALL problems here. This is an extremely good basic to advanced topic-wise problem set.
- Blog of Codeforces blogs
- USACO Guide - It also covers some very advanced topics in CP and has a pretty neat interface.
- AtCoder DP Contest - The best collection of all forms of DP problems.
- It is impossible to mention all the resources for CP. The critical thing is always to keep practicing and participating in contests. Say you solved 3 problems in a contest, then see the 4th problem. If it involves some idea or algorithm you did not know, then learn it. That's how you improve.

Summers before CDC

- GOC CDC Series - A very important collection of past problems from companies' coding tests. (To get all contests, keep changing the number at the end of the URL, GOC FB Page)
- Interviewbit - Interview-type problems, try to solve all of these, cover a lot of things and aspects.
- LeetCode - Interview-type problems, and organize weekly and bi-weekly contests.
- Leetcode top interview questions - Easy
- Leetcode top interview questions - Medium
- Leetcode top interview questions - Hard
- Leetcode top interview questions - Compiled
- 100 most liked questions on Leetcode

OOPS

PPD's Software Engineering course is the best resource you will possibly get for this.
Slides
Video Lectures

Software/Systems Roles at HFTs

These roles mainly involve low-latency development in C++, and the interviews focus on core CS concepts like OOPS, computer architecture, operating systems, networks, and in-depth knowledge of the C++ language. Tests for these roles include CP problems, MCQs on systems knowledge, and even some questions like predicting the output of several code snippets, which require a very good understanding of C++, system calls, and core systems concepts. Interviews generally don't focus a lot on CP (for dual-degree people in internships/placements and B.Tech people in placements). Instead, they comprise questions on core systems ideas and C++ knowledge and features. However, for B.Tech people during internships, the interviews can mainly be focused on CP problems, OOPS, and C++ knowledge.

OOPS

- PPD's Software Engineering course again
  Slides
  Video Lectures (upto lecture 21)

Computer Architecture

Resources

- COA - Autumn 2021 or COA - ETH Zurich
- HPCA Georgia Tech - Part 1, Part 2, Part 3, Part 4, Part 5, Part 6 or HPCA - Spring 2020-2021 (don't need to study everything, study mainly the topics mentioned below)

Important Topics

- Caching (various cache types, cache organizations, cache coherence, PIPT caches, aliasing in VIPT caches)
- Pipelining
- Branch Prediction (taught in HPCA)
- Out of Order Execution (taught in HPCA)

Operating Systems

Resources

- OS - UC Berkeley (make sure to watch this; extremely good and THE BEST)

Important Topics

- Process Basics
- System Calls
- Process Synchronization, Deadlocks
- Multithreading
- Virtual Memory, Paging

Computer Networks

Resources

- Networks - Prof. AG - Spring 2022 or Networks - Northwestern

Important Topics

- Some questions focus on application layer protocols like DNS and DHCP.
- And the transport layer is the most important. You should know everything about TCP and UDP, everything!
- Questions on the network layer are sometimes asked, too - IP.
- Socket programming details - system calls, etc.
- **Please note that these are only the most asked topics, but you should have in-depth knowledge of each layer of the network stack.**

C++

Resources

- Cherno - C++ (watch it ALL)
- CPP Reference
- PPD Slides
- Some HFT Questions
- C++ Interview Questions
- Template Metaprogramming (first 2 videos)

Important Topics

- Implementation of STL containers
- **Everything** in PPD's slides
- Smart Pointers (unique_ptr, shared_ptr, how they can be implemented)
- Move Semantics
- Templates
- Template Metaprogramming
- In short, there are many things you would be using while writing a C++ program. Say, for example, you are using the new keyword, then you should know what it actually does under the hood, what system call it makes, etc. Basically, you should know how every abstract thing is implemented under the hood.

Quant

These roles are offered by high-frequency trading firms, investment managers, and investment banks. The job involves researching, analyzing, testing, and optimizing trading strategies.

How to Get an Interview Shortlist?

Along with a good CGPA, having prior mathematics olympiad and/or competitive programming achievements goes a long way in getting a shortlist, as these are the highlights recruiters look at the CVs of candidates. Prior internships or experience in the field are usually not required and rarely help get a shortlist. Most firms also conduct tests involving solving programming problems and puzzles.

Courses

As a prerequisite, studying the following courses should be enough:

Probability and Statistics

- Stat 110 - Harvard (Notes) (Problems)
- Book
- Statistics - StatQuest
- Essence of Probability - 3B1B

Discrete Structures (focus on propositions, proofs, induction, and counting)

- Discrete Structures - MIT
- Problems - MIT
- Tutorials - IIT Kgp

Calculus

- 3B1B - Calculus
- Brush up on very basic differentiation/integration

There are also some topics, questions related to which are not asked in CDC interviews but are common in off-campus interviews. These can be treated as non-essential topics that can be covered with extra time.

Linear Algebra

- 3B1B - Linear Algebra
- Linear Algebra - MIT

Machine Learning

- Machine Learning - Stanford
- Machine Learning - StatQuest

Resources

The interviews and tests in this profile are mainly focused on evaluating a candidate's aptitude in mathematics and structured problem-solving skills. The problems mainly revolve around probability, statistics, logical puzzles, induction, basic calculus, etc.

After learning this, it is mostly about practicing as much as you can. Numerous problem sets are available, so a problem set rarely limits the amount of practice. Apart from doing many questions, focus on absorbing the generalized problem-solving skill that is required in solving the questions you have practiced.

Essentials

These are the resources that cover almost every kind of question that might be asked in the interview. Mastering these should be enough to crack most on-campus interviews.

- Problem Solving - A typical quant
- Heard on the Street by Timothy Falcon (Solve sections on puzzles/probability)
- 50 Challenging Problems in Probability by Frederick Mosteller
- Brainsteller
- PuzzledQuant (Closest to actual interview questions)
- Puzzles and Proofs - 3B1B (good videos to learn problem-solving)
- Game of NIM and Grundy Numbers (Variations of NIM usually asked in interviews)

For Further Practice

If you still have some time left and want to practice more, you can pick and choose from the given list according to your liking.

- A Practical Guide to Quantitative Finance Interviews by Xinfeng Zhou (Easy)
- http://puzzles.nigelcoldwell.co.uk/ (Easy-Medium)
- https://gurmeet.net/puzzles/ (Easy-Medium)
- https://pratikpoddarcse.blogspot.com/ (Medium-Hard)
- TED-Ed Riddles (Videos)
- Probability Puzzles - Presh Talwalkar (Videos)
- Easy ~ Medium Puzzles - Presh Talwalkar (Videos)
- Presh Talwalkar - Math Puzzles - Volume 1
- Presh Talwalkar - Math Puzzles - Volume 2
- Presh Talwalkar - Math Puzzles - Volume 3
- The Puzzle Toad - CMU (Hard)
- https://www.braingle.com/brainteasers/Probability.html
- https://www.cs.cmu.edu/puzzle/index.html
- https://www.ocf.berkeley.edu/~wwu/riddles/easy.shtml
- http://www.qbyte.org/puzzles/
- CMU Game Theory

Fast Math

Increasing your calculation speed also helps a lot in interviews. Some companies have also asked such questions directly to measure calculation speed.

- https://arithmetic.zetamac.com/
- https://www.mathtrainer.io/

Blogs for Interview Experiences

When targeting specific profiles or companies, you can get a lot of wisdom by learning from people who have already gone through the process before successfully. CodeClub releases the blogs of such people from time to time, you can go through them at the provided links.

- Rubrik - Ashutosh
- Quantbox - Suhas
- Internship Blogs Archive
- Placement Blogs Archive