

b'

Google Interview Preparation For Software Engineer \xe2\x80\x93 A Complete Guide

- Difficulty Level : \nExpert
- Last Updated : \n17 Jun, 2021

Google \xe2\x80\xa6 The top attractive employer and one of the highest paying companies in the world. Google opens a door for exceptional programmers who are not limited to solve problems they already know but also they should have skills to solve problems they haven\xe2\x80\x99t encountered before. If you are one of them who has this skill and if you have a dream to land up a software engineer job role in Google then you need to follow a strategy and you need to understand the complete process that this company is looking for the hiring of top-notch programmers.\xc2\xa0

\xc2\xa0



Hiring procedures in Google for the software engineering role is similar to other companies so we will discuss the programming skill that you must have and we will also discuss some other tips and details that you should keep in mind while applying at Google.\xc2\xa0

\xc2\xa0

Important Points

\xc2\xa0

- Google hires only exceptional programmers so there is no doubt that problem-solving and coding skill (Focus area data Structures and Algorithms) is a must-have skill in Google for software engineering role but you need to keep in mind that Google also care a lot about [Googlyness](#) that covers passion for technology, curiosity, ethics, friendliness, good citizenship, and more.

- Difficulty level of the interview depends on the level of software engineering role you are applying in Google. Software Engineer or SWE-II (Level 3) is an entry-level full-time software engineer. At this level, there are 4 or 5 onsite rounds and people on the cusp of L3 and L4 (below), they may throw in a design question, but usually not. SWE-III (Level 4) is for candidates with BS + 8 years, MS + 5 years, Ph.D. + 2 years. Approximately. At this level, expect 4 or 5 onsite rounds also at least one Systems Design question.
- Check [List of skills](#) that *Google in Education Team* has released which they expect from a potential engineer.
- All engineers (at every level) collaborate throughout the Google codebase, with an efficient code review process and that's the reason Google is keen to see really high quality, efficient, clear code without typing mistakes.
- Interviewers are trained to not react to your answers so don't expect yes or no answer from them that if you are heading in the right direction or not. This sucks but is a very effective way of assessing a candidate so prepare yourself to see stone-cold faces.

Before we discuss all the details we highly recommend you to take a look on [How we hire](#) on Google career page and get familiar with interview process. For Google software engineering interview you can prefer any programming language you are comfortable with but you should know your programming language very well, It would be great if the choice is [C++](#) or [Java](#). Let's discuss all the rounds one by one from getting the call for the interviews to cracking the interview in detail.

Getting Noticed by Google

Google receives more than two million resumes but hires only around *4000 employees every year* so it's not easy to get noticed by recruiters for interview calls. Take the referrals through some connections working in these companies or you can take the help of LinkedIn and online coding platforms to increase your chances for interview calls. Below are some points that will help you in getting the call for interviews.

- Try to make a single-page resume and include only relevant things. Make it short, precise because none of the recruiters has more than 15 seconds to scan a CV. Write down only those things you are comfortable with, do not fake anything because the interviewer can spot that easily during the interview. If you are mentioning your project then mention the complexity of your project as well. Check the link [Resume Building Resources and Tips](#) for the guidelines to prepare your CV.
- You can participate in [Google Kickstart](#) or [Google Code Jam](#) which is a coding competition hosted by Google. If you reach Round 2 (around 3000 contestants) in Google Code Jam, you might be contacted by Google.
- Keep your LinkedIn, Github and other profiles updated, hiring managers uses these tools like LinkedIn, alumni databases, and professional associations to find out suitable candidate.
- Check the link [Create Your Resume for Google: Tips and Advice](#)
- Go to conferences and start networking. Google employees often speak at conferences and they are very approachable.

Telephonic Interview

Once your resume will be shortlisted you will be invited for the technical phone interviews (Average 2 phone interviews) which are slightly different than onsite interviews but there you need to write code on some sharing tools like Google Doc and that will be visible to the interviewer. It will be for around 45 minutes covering [Data Structures](#) and [Algorithms](#). You may encounter two questions and you are expected to write code for at least one of them. Below are some points to guide you for phone interviews.

- You need to write the code for the coding question which interviewer will ask, throw out a brute force solution in the beginning, but you should try to improve your solution as well. Think about the time and space complexity and see if you can improve those.
- Practice writing code on Google Doc. It is different than a text editor. There's no indentation,

syntax highlight and auto-completion so it's good to practice on Google doc and get familiar with it.

- **Think out loud** and speak clearly (people face trouble here) in phone interviews and tell your strategy to the interviewer to solve the problem.
- The document will update in real-time. It's going to be obvious if you are typing your answer or if you are copying and pasting it.
- The interviewer may ask to do second phone interview if in the first one they won't be able to assess you properly.

Onsite Interview

After the phone interviews, Google conducts 4-6 onsite interviews including lunch interviews where lunch interview isn't a real interview and it's just the interaction with Googler. Each round roughly takes 45 minutes to one hour and this will be a whiteboard round and the majority of the Google interview consists of coding. Now here comes the role of DSA and Core CS subjects to check your coding, problem-solving skill and command over computer science fundamentals. Firstly you will go through the coding interview round and then system design (based on experience) round. The difficulty level of the question is based on the level you are applying for in Google.

1. Technical Coding Round and General Analysis Questions

- The candidate is tested on whether he/she can solve real-world problems using the knowledge of data structures and algorithms. The expectation is to be able to arrive at the most optimized solution to the given problem. Your ability to convey your algorithmic knowledge along with code is must.
- In general analysis question interviewer will give you a mathematical, design, or opinion-based question where they will investigate your thought process and how you would proceed as an employee.
- We highly recommend you to go through CTCI ([Cracking the Coding Interview](#)) book, practice questions specially on [GeeksforGeeks](#), [Leetcode](#) and [Programming Interview Questions | CareerCup](#) for Google interview preparation. You can also practice for the same on [Google Preparation](#).
- Below is the distribution of the types of problems that were generally encountered in Google interview. This data is based on the interview experiences from Glassdoor for Google.

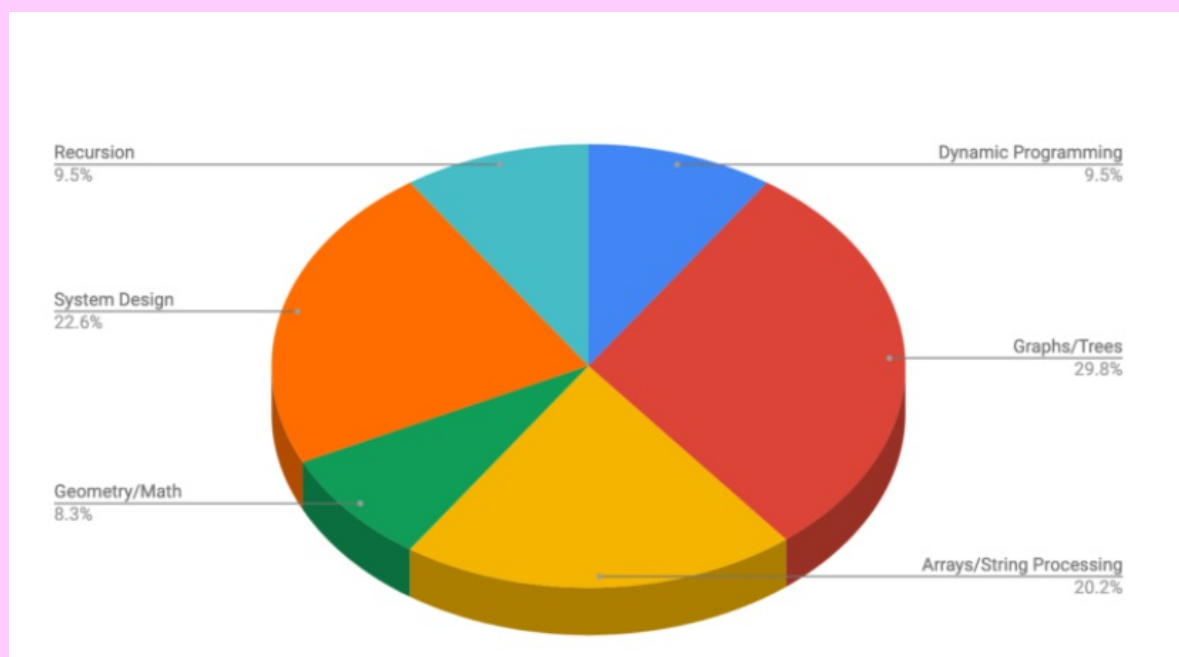


Image Source: [byte-by-byte.com](#)

- Check the link [How to Get a Job in Product Based Companies?](#) to deal with in-depth technical coding question, what approach and tips you should follow during the interview.

- Don't do bubble-sort. You should know the details of at least one $n \log(n)$ sorting algorithm, preferably two (say, quicksort and merge sort). Merge sort can be highly useful in situations where quicksort is impractical, so prepare yourself with that.
- Go through the basic tree construction, traversal, and manipulation algorithms. Get familiar with the topic binary trees, n-ary trees, and trie-trees, at least one type of balanced binary tree, for example, red/black tree, a splay tree or an AVL tree along with the implementation. Understand tree traversal algorithms: BFS and DFS, inorder, postorder and preorder traversal.
- Get familiar with how to represent a graph in memory, basic graph traversal algorithms. Do not focus too many fancy algorithms like Dijkstra. Study this topic if you have enough time.
- You should know about the most famous classes of NP-complete problems, such as traveling salesman and the knapsack problem, and be able to recognize them when an interviewer asks you them in disguise.
- The interviewer may ask some basic discrete math questions so spend some time there as well and take a look on combinatorics and probability also get familiar with n-choose-k problems.
- Don't forget about one of the most important CS subject Operating System. Threads, concurrency, processes, deadlock, livelock and read about the other topics in OS.
- Interviewer will definitely check that how you decide which data structure is suitable for the problem you have given by your interviewer.
- Do not forget to test your code with all the cases. After writing the solution you have to test your code, even if minimally because if you don't do that, points will be deducted.
- One of the mock interview video featuring real Google engineers [Example Coding/Engineering Interview](#) will definitely help you in solving coding related question .

2. System Design Round:

- **System design questions** is an open-ended conversation and this round is mostly conducted with an experienced candidate where they test your overall ability to design and scale technically based systems like design *Gmail*, *youtube*, *uber* etc.
- Check the link [System Design Interview Question](#), [System Design Primer](#).
- L4 and below can get OO design and maybe some systems design-related question. Here you're expected to know the high-level details. If you are applying for L5 and up you're going to be asked systems design questions, with the expectation that you can define the goals/requirements of the system, define the high-level architecture, deep-dive into some components, and identify the bottlenecks.
- In Google, scalability is always important. So expect design questions that will be an ambiguous real-world problem for large size systems. They are looking for the process of thought and how you break things down to ultimately get too creative and scalable solutions.

What Interviewers Are Looking For?

We are not simply looking for engineers to solve the problems they already know the answers to; we are interested in engineers who can work out the answers to questions they had not come across before.

The above thing is the most important thing the interviewer will be looking for also read the below points

- Interviewer have generally a mindset while taking the interview that if he/she can work along with the candidate in his/her daily to a job or not so do not show signs of arrogance or ego while giving the interviews.
- Interviewers will check whether you've used the suitable data structures and algorithms while implementing the code or not.
- Interviewers will notice the approach that how you optimized the solution, your knowledge about the choice of programming language, your coding speed, any corner cases that you missed and how you analyzed time and space complexity.
- They will check how you communicated your thought process to solve a specific problem in a logical way. They will also check if you were able to catch the hint and was able to proceed with the solution or not.

- If the candidate was open to new ideas? if the candidate was flexible in his/her solutions?

Interview Evaluation Criteria: All the feedback is collected from different interviewer after the interview and then you are rated on a scale of 1-4 in a bunch of different categories that include your coding experience and analytical skill. This feedback is then sent to a hiring committee for the final decision.

Tips:

- Learn to **think out loud** during your whole interview otherwise interviewer will have no clue what you are thinking. Show your interviewer your thought process about the problem and what approach you are going to follow to solve the problem.
- Practice by writing code on paper or whiteboard. It will really help you during your interview.
- Never say that you cannot do it. Even if there is a problem that you have not solved before or seemed that you can't solve it, keep attacking the problem from different angles, the interviewer will give you hints. But if you would say that you can't solve the problem it is a big red flag and you might end up getting rejected.
- Always ask relevant questions before jumping to the solution to make it more clear. Always check your assumption with the interviewer and clear all your doubts before heading to the solution.
- We highly recommend you to not write pseudo-code to design your code. You don't have time for that in a 45-minute interview.

Helpful Links:

- [Google Interview Preparation Questions](#)
- [Google Interview Experience](#)
- [Practice for Cracking Any Coding Interview](#)
- [Glassdoor Google Software Engineer Interview Questions](#)

My Personal Notes

Add your personal notes here

Save

,