Cracking the Google coding interview: The definitive prep guide

Passing the Google interview is any software developer's dream. This interview is the ultimate test of your technical prowess and requires not luck or shortcuts but hard work and preparation. Luckily, Google is very transparent about their expectations for candidates. The more you know about Google's process and company culture, the more likely you are to succeed. So today, we'll take a deep dive into their coding and behavioral interview process to show you how to crack the interview.

Today we will go over the following:

- Interview Overview and FAQ
- Google Interview Process
- Coding Interview Question Guide
- Behavioral Interview Question Guide
- Resource Guide

Overview of Google interviews

The whole interview process takes 2 to 2.5 months to complete. Google interviews for software developers consist of 5 to 7 interviews in total. This includes an initial phone screen with a Google recruiter, followed by 4 to 6 on-site interviews with different Google employees. All coding challenges are done on a whiteboard, Google Docs, or a Chromebook.

Google normally hires at level T3 with T11 being the height of levels. T3 is an entry-level, full-time software engineer with an undergraduate or Master's degree. The average salary for a T3 employee is $117,000 annually. Google prefers the following programming languages: Java, C++, C Go, and Python.

There are three types of coding problems you can expect to see in a Google interview.

- **System design questions:** these questions gauge your ability to handle high-level system design with scalability in mind.
- **Coding interview challenges:** these questions gauge your knowledge of data structures and algorithms to optimize a solution to common problems.
- **General analysis questions:** these questions gauge your thought process through mathematical or opinion-based questions

What is unique about Google interviews?

Though software engineer interviews at Google are similar to other big tech companies, there are some unique aspects of their process. So, unlike Apple, Microsoft, or Amazon, what's different?

Clear expectations: Google is not shy about their interview process, so there is little guesswork when it comes to preparation or expectations. Google provides all the details of what to expect on their "careers" page.

Coding in Google Docs: During the initial phone screen, you will be expected to code in Google Docs. It's important to study how to code well on this platform, as it can take some practice.

Hiring committee: Google aims to review applicants without bias, so they utilize a hiring committee for all candidates.

Grading scale: Each interviewer is graded on a scale of 1-4 based on the four hiring criteria where 3 is the threshold for hire vs. no hire.

Four hiring criteria: Google focuses on cognitive ability, "Googleyness", leadership skills, and technical skills in their candidate review process. Let's break that down in the next section.

What does Google look for in a candidate?

Cognitive ability. General cognitive ability refers to your problem-solving skills, abstract thinking, curiosity, and willingness to learn. Google looks for smart people who can think complexly about themselves, their teams, and their projects.

Googleyness. Like all organizations, Google has a specific company culture, and they look for candidates who mesh well with their values. Google's main focus is on people; they care about inclusivity and the improvement of human life while remaining ethical. Google is known for small teams and a laidback environment to encourage creativity, innovation, and open communication.

Leadership skills. Google hires candidates with emergent leadership skills. For Google, leadership is the ability to step in and out of difficult problems when you are needed while promoting ethics and safety. Google also looks for those who empower others to self-organize and foster open communication.

Technical skills. Google hires candidates with the strongest coding abilities, and they assess technical skills mostly on *conceptual* understanding, not memorization. They assess coding skills on the following topics:

- Algorithms

- Sorting
- Data structures
- Graphs
- Recursion
- Object-Oriented Programming
- Big-O Notation
- APIs
- How to test your code
- Mathematics (*such as n-choose-k programs*)

Step 1: Before the Interview

Preparation. Preparing for a coding interview is a lengthy process. I encourage three months of preparation in advance. You need a preparation plan to stay on track. Look at our 3 Month Coding Interview Preparation Bootcamp for a plan created by real hiring managers. Be sure to prepare using a language that Google prefers, such as C++, Java, Python, Go, or C.

Update your resume. Your resume needs to catch the attention of a Google recruiter in *six seconds or less*. One of the most common reasons talented applicants don't get an interview is due to a poorly crafted resume. Spend some time updating your resume.

If your resume passes the test, they'll schedule a call to learn more about your skills and experience. It will be about a week before you hear from the recruiter.

Step 2: Prescreen with Google Employee

Your phone screen will last between 45 - 60 minutes, likely on Google Hangouts. The Google employee will test you with coding questions related to data structures and algorithms. You will solve these on a Google Doc, using around 20-30 lines of code.

It is important to communicate your thought process as you work; this is how they gauge your general cognitive ability.

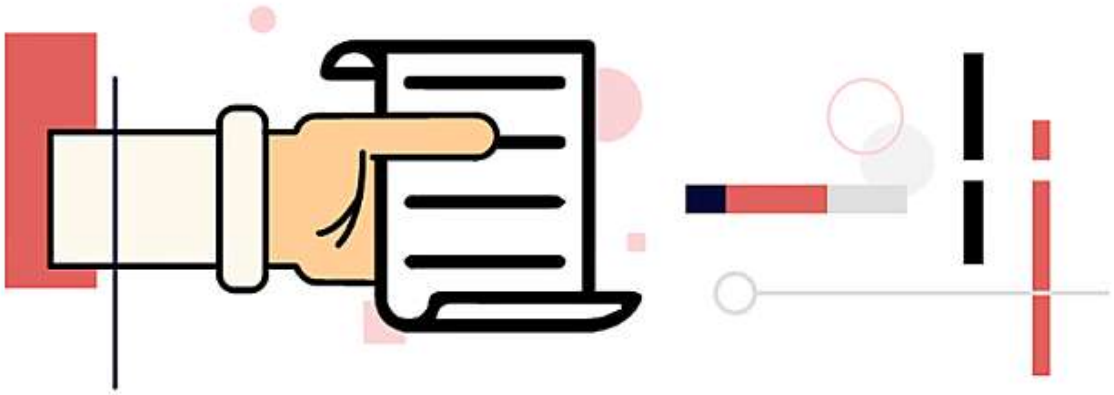
You can expect an open-ended coding challenge. Asking further questions for clarification is a great way to demonstrate problem-solving skills. If you finish before the time ends, look for ways to optimize and check for bugs. Always be sure to consider corner and edge cases.

On-site interviews

If you pass the prescreen, you will be invited to an on-site interview. You will meet with 4-6 Google employees for 45 minutes each.

These onsite interviews will heavily focus on coding, focusing on data structures and algorithms. You will be writing code either on a whiteboard or a Chromebook, which they will provide. It's a good idea to ask the recruiter beforehand so you can practice properly.

The on-site interviews also feature behavioral interview questions to assess who you are as a potential employee. Google wants to see that you fit with their company values, so be sure to prepare for behavioral interviews as well. We'll discuss this more later.



The offer / no offer

You will be scored on a scale of 1-4, with 3 being the threshold of hire or no-hire. Once your interviews are complete, the feedback will be sent to a hiring committee to make the final decision. This process can take several weeks, and it's okay to send a friendly nudge email if too much time has passed.

If you get an offer, be ready to discuss salary and start date, so it's a good idea to brush up on negotiation skills. If you do not get the offer, you may have to wait six months to reapply, which gives you plenty of time to prepare and study up on your weak points.

Practicing for coding questions takes a lot of time, effort, and focus. Let's break down the top Google coding questions as well as actionable advice to prepare.

We will outline the top 15 coding questions to get you familiar with the kinds of questions expected of you. To see detailed answers, check out part 2 of this series, Google Coding Interview Questions: top 18 questions explained.

Top 15 Google coding interview questions

Find the kth largest element in a number stream

Problem Statement: Design a class to efficiently find the Kth largest element in a stream of numbers. The class should have the following two things:

- The constructor of the class should accept an integer array containing initial numbers from the stream and an integer 'K'.
- The class should expose a function add(int num) which will store the given number and return the Kth largest number.

Find ‘k’ closest numbers

Problem Statement: Given a sorted number array and two integers ‘K’ and ‘X’, find ‘K’ closest numbers to ‘X’ in the array. Return the numbers in the sorted order. ‘X’ is not necessarily present in the array.

Delete node with given key

Problem statement: You are given the head of a linked list and a key. You have to delete the node that contains this given key.

Copy linked list with arbitrary pointer

Problem statement: You are given a linked list where the node has two pointers. The first is the regular ‘next’ pointer. The second pointer is called ‘arbitrary_pointer’ and it can point to any node in the linked list.

Your job is to write code to make a deep copy of the given linked list. Here, deep copy means that any operations on the original list (inserting, modifying and removing) should not affect the copied list.

Mirror binary trees

Problem statement: Given the root node of a binary tree, swap the ‘left’ and ‘right’ children for each node.

Find all paths for a sum

Problem statement: Given a binary tree and a number ‘S’, find all paths from root-to-leaf such that the sum of all the node values of each path equals ‘S’.

Longest substring with no more than ‘k’ distinct characters

Problem statement: Given a string, find the length of the longest substring in it with no more than K distinct characters.

Longest substring with no repeating characters

Problem statement: Given a string, find if its letters can be rearranged in such a way that no two same characters come next to each other.

Equal subset sum partition

Problem statement: Given a set of positive numbers, find if we can partition it into two subsets such that the sum of elements in both subsets is equal.

Determine if the number is valid

Problem statement: Given an input string, determine if it makes a valid number or not. For simplicity, assume that white spaces are not present in the input.

Print balanced brace combinations

Problem statement: Print all braces combinations for a given value ‘N’ so that they are balanced.

Given a number of tasks, determine if they can all be scheduled

Problem statement: There are ‘N’ tasks, labeled from ‘0’ to ‘N-1’. Each task can have some prerequisite tasks which need to be completed before it can be scheduled. Given the number of tasks and a list of prerequisite pairs, find out if it is possible to schedule all the tasks.

Implement a LRU cache

Problem statement: Least Recently Used (LRU) is a common caching strategy. It defines the policy to evict elements from the cache to make room for new

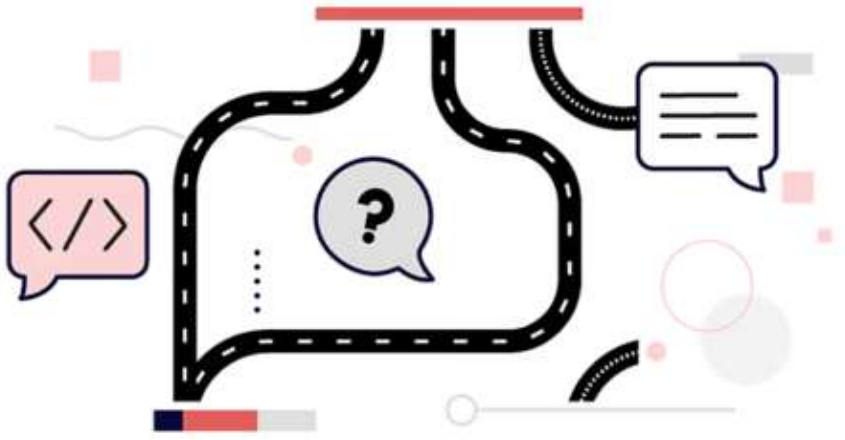
elements when the cache is full, meaning it discards the least recently used items first.

Find the high and low index

Problem statement: Given a sorted array of integers, return the low and high index of the given key. Return -1 if not found. The array length can be in the millions with many duplicates.

Merge overlapping intervals

Problem statement: You are given an array (list) of interval pairs as input where each interval has a start and end timestamp. The input array is sorted by starting timestamps. You are required to merge overlapping intervals and return output array (list).



12-week preparation roadmap

Preparing for a Google coding interview is strategic. It requires months of prep and practice to master the right concepts and develop confidence. Let's look at the definitive 12-week prep plan proven to help candidates land jobs at big companies.

Week 0. Choose a programming language based on Google's expectations and your preferences.

Week 1. Review the basics of your programming language. If you brush up on the basics, you're less likely to stumble during your interview. Review concepts like how to read input from the console; how to declare and use 2D arrays.

Week 2 & 3. Familiarize yourself with data structures and algorithms. These are essential to coding interviews with Google.

Data structures you should know:

- Arrays
- Linked Lists
- Stacks
- Queues
- Trees
- Graphs
- Heaps
- Hash sets
- Hash tables/maps

Algorithms you should know:

- Breadth-first search
- Depth-first search
- Binary search
- Quicksort
- Mergesort
- A*
- Dynamic programming
- Divide and conquer

Week 4 & 5. Practice data structure and algorithmic challenges with sites like Educative or Leetcode. Start practicing simple coding problems. This will make it easier down the line to tackle harder questions.

Weeks 6-8. Practice complex coding problems, and start timing yourself. It's important to consider Runtime and Memory complexity for each solution. For practice and automated challenges along with interactive solutions, look at Grokking the Coding Interview: Patterns for Coding Questions

Weeks 9 & 10. Study the top 10 System Design Interview questions. These are now an integral part of the interview process and impact your *hiring level*.

Week 11. Study OS and Concurrency concepts. These questions are used to gauge your hiring level. Brush up on multithreading fundamentals to stand out for higher levels in Google's ladder.

Week 12. Study object-oriented programming and design questions. These questions gauge your critical thinking, project-based problem-solving skills.

Tips for practicing coding challenges

There is no shortcut or magic wand for practicing coding challenges. Here are some basic tips to guide you through the preparation stage.

Keep time in mind. The coding interview will be timed, so it's important to prepare with that in mind. If you are used to preparing under a time constraint, it will be far less stressful during the actual interview.

Know your weak spots. As you prepare, take note of your weak spots. Everyone has them. Google has stated that they care about your thought process, so if you come up against a weak spot, talk through it. This will demonstrate your eagerness to improve.

Know the common pitfalls. There are three big pitfalls when it comes to a Google interview: not knowing the Big-O complexity of an algorithm, having no knowledge of Google's expectations, and not articulating your problem-solving process. Keep these pitfalls in mind as you work.

Articulate your process. Google wants to hear about your thought process. As you practice, get used to explaining why and what you are doing. Those with a clear sense of how they work stand out.

Behavioral interviews are often overlooked by software development candidates. In reality, this is the interview that sometimes will make or break you as a candidate. Google cares deeply about their values, so if you come unprepared for these questions, they'll notice it.

Cultural and behavioral interviews are there to weed out the people who may have the skills but don't have the mentality. Let's dive into Google behavioral interviews.

What to expect from a Google behavioral interview

Behavioral interviews at Google test how you act in employment-related situations or conflicts, both positive and negative. Behavioral interviews help an employer decide if you're someone they want to work with. These interviews will ask you to reflect on your past performance and behaviors to gain a sense of how you act under pressure and how you understand professionalism. You can expect three types of questions:

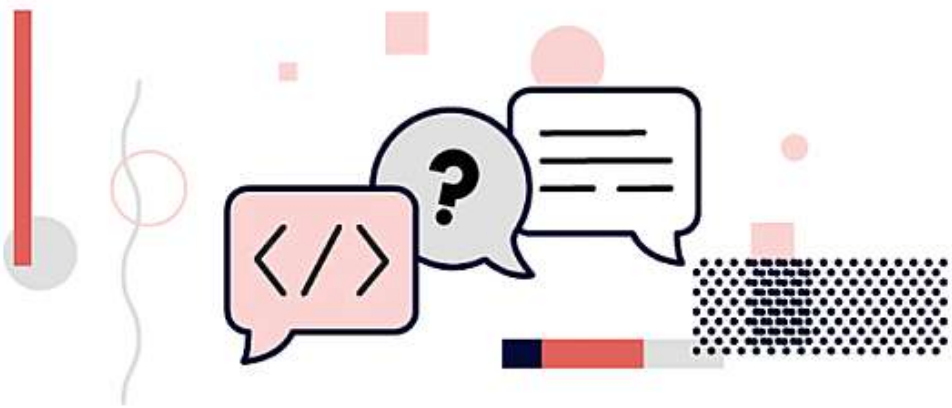
- Past experiences
- Hypothetical situations
- Values-based

How to prepare for behavioral interviews

Google has outlined 8 steps for preparing for behavioral interview questions. Study these and use them to shape your preparation and answers.

- **Predict the future.** You can anticipate most of the questions you will be asked. You have the resources to prepare your answers. See below for 10 common questions.
- **Plan.** Write your answers down. Practice strategically. Don't wing the behavioral questions. Think back to your work as an intern or other past experiences. Write down specific examples or notable accomplishments.
- **Have a backup plan.** Google recommends having 3 answers per question. This helps you prepare diverse, interesting answers.
- **Explain.** Google asks you to explain your thought process and decision-making. Explicitly stating your assumptions and processes helps you stand out.

- **Be data-driven.** Google wants answers that relate directly to tangible growth, change, or demonstration of skill.
- **Clarify.** You can use open-ended questions to offer insight into your value as a candidate.
- **Improve.** Google encourages you to always focus on improvement. You can start with a brute force answer, but then work through how you could improve your process.
- **Practice.** Google encourages you to practice aloud to construct clearer answers. Ask a friend to conduct a mock interview or record yourself answering questions.



10 common behavioral interview questions

Google has compiled their own list of behavioral interview questions that their hiring managers use in interviews.

Let's go through them.

Describe your work process for _____.

While there are no wrong answers to this question, it can be used to gauge how your experience impacts your day-to-day life as a developer. A good answer dives into your workflow process, tools, and ability to work on a team.

Tell me about a time when you took a risk and failed.

This question provides insight into how you learn and approach difficulty. A good answer demonstrates honesty and active learning in the face of failure.

What is an accomplishment that you're proud of?

This question is used for building diverse teams. Google is looking for your specialties, sense of success, and your values in the long term. A good answer looks beyond the accomplishment and prods at what it means about you as a person.

What is the most complex thing you know a lot about? Teach me about it.

This question looks at your skills, communication skills, and ability to explain complexity to others. A good answer focuses less on information and more on the effectiveness of your teaching style.

If you join, how will you impact your team?

This question looks at your ability to fit with others in a structure. A good answer shows that you did research on Google's values. Top answers speak to Google's goals.

Tell me about an unstructured environment you've worked on.

This question gauges your work style by defining what unstructured means to you. A good answer relates your workstyle to your position. Are you able to fit into a predefined structure?

What do you want me to know about you that we haven't discussed?

Good answers show extra skills you have that may be beneficial to Google. They will also look for soft skills to see if you have non-technical experiences that benefit their teams.

What is your favorite Google tool, and how would you improve it?

This question gauges your knowledge of Google, creativity, and willingness to affect change. A good answer is data-driven but not too technical; be sure to focus on why you want to change something. Remember: it should make people's lives easier.

What does "being Googley" mean to you?

This question is a chance to show off your cultural fit and values. A good answer avoids too much jargon but speaks to the underlying values of Google's culture.

What scares you?

This question looks to see your weak spots and stressors. A good answer is self-reflective and aims not to simply overcome your fears but to understand how they impact you as a worker.

Part 5: Resource Guide

Now you have a good sense of what Google interviews are, how they differ from other companies, and how to prepare.

I've compiled a list of all the resources you need to study, practice, and build the confidence needed to excel in a Google interview. Check out part 2 to start learning coding interview questions!

Happy learning!