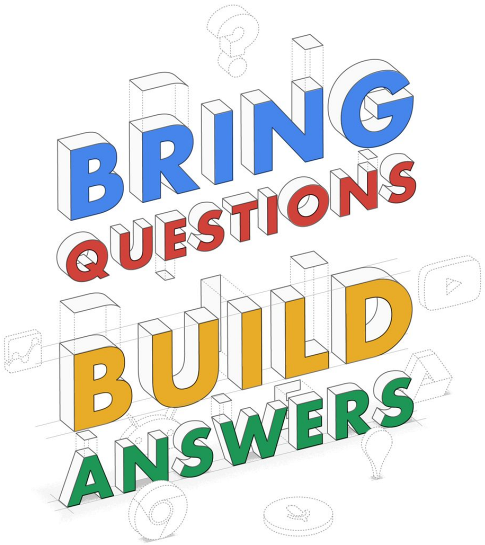

# Google On Site Interview Preparation

Technical Solutions Consultant, gTech

*A reminder that all of the information in this document is confidential. Please do not to share this with anybody else.*

# Technical Solutions Consultant - *On Site Interviews*

**Congratulations on passing the hangout interview! We hope you enjoyed it. For the next stage, you will interview with different members of the gTech team, so expect more detailed discussions. We also hope to give you an insight into working at Google, so relax, enjoy the experience and of course ask any questions of interest!**

## What to Expect:

A typical on site interview process will contain 4 different interviews. These may be split up or be spread across multiple days, or all on one day, depending on your current location/availability. We may also have to have these over Google Hangouts again, if we are not able to have them face to face.

On Site Interview format:

1. Interview with the Hiring Manager
2. Interview with a gTech Software Engineer
3. Interview with a Peer (somebody from the Team you will be joining)
4. Interview with a Cross Functional Peer (member of the wider gTech team)

**PLEASE NOTE:** You will need a computer with Internet access for the engineering interview to write code using a shared Google Doc. Please let us know if you would like to bring your own laptop to the interview (if the interview is being held in our office), or whether you would like to use one of ours.

## On Site - Hiring Manager Interview

### What to Expect:

This interview will focus mainly on your Leadership qualities and 'How You Think'. The Manager will want to check you are a good fit for the role specifically, and also test your ability to think on your feet.

1. **Leadership** - Experience leading projects and/or teams. Showing thought leadership, initiative and influencing cross functional/senior stakeholders without official authority. Leadership at Google is not always just about being able to manage a team, but to be able to show leadership skills in general, or in projects and in a team.

2. **How You Think/Decision Making** - Questions designed to see how you think on your feet and your approach to dealing with complexity and ambiguity.

3. **Project Management** - Ability to manage projects and time scales. Working with cross functional teams and external clients.

4. **Decision Making** - Ability to use data in your decision making and to make difficult decisions. How do you make decisions and what do you use to make informed decisions (i.e data, consultation etc).

### Tips

- The Hiring Manager wants to know you are the best person for this job. Can they envisage working and managing you? Do you show the initiative and ability to think on your feet?

- Don't be afraid to ask questions! Ensure you clarify questions you don't fully understand, and ask for more data if you need it. Make assumptions if you need to, but make them out loud.

- We aren't always looking at the end result. Often, we are trying to see the steps you take to get to a solution, the reasoning and decisions you make (essentially, the journey is just as important as the ending).

## On Site (Peer) Interview

### What to Expect:

This interview will be fairly technical, and will dig deep into your Role Related Knowledge. Expect to be tested on the following;

1. **Web Tech** - This will be more in-depth that the telephone interview, looking at knowledge of front end tech, how they differ from each other and possibly asking more hands-on questions.
2. **Troubleshooting/Problem Solving** - giving you scenarios and seeing the steps you take to solve a problem.
3. **Databases** - Knowledge of SQL/ Queries or any other Database.
4. **Analytical Skills** - Ability to analyse data and present it back to an audience. Experience of doing this.
5. **Unix/Linux -** Knowledge of commands and general use of *nix systems.
6. **Google Products** - What are your favourite products, the pro's and con's of them. Knowledge of Advertising (AdSense, Doubleclick, AdEx and AdWords) and how the advertising ecosystem works.

### Tips

- Be detailed but concise in your answers. This is especially the case for your knowledge of Web Tech - try to avoid only superficial answers and having to be prompted.
- Read up on our products. Check with your recruiter whether you should look more at Consumer products or Advertising Products, and if they can suggest any that might come up. Definitely grasp the basics, like the difference between AdWords and AdSense.

# Technical Solutions Consultant - *On Site Interviews*

## On Site (Cross Functional Peer) Interview

### What to Expect:

This will focus more on your non-technical skills and culture fit ('Googleyness'). Expect questions on all/some of the following;

1. **Client Facing ability** - How you work with clients, different projects and how to manage client expectations.

2. **Cross Team Collaboration** - How well you would work with the sales teams. Previous experience of this and being the technical face of a company.

3. **Communication** - How you explain technical things to non technical people. How you tailor communication when speaking with Engineering vs. Product vs. Sales, etc.

4. **Knowledge of Google** - Our products, how we operate (and generate revenue), our core values and our place in the industry.

### Tips

- Your client facing ability is key in this interview. The interviewer wants to see somebody they would be comfortable putting in front of our clients and sales teams.

- The interviewer will be keen on seeing somebody adept at both pre- and post-sales work.

- Ensure you are not using technical lingo if they ask you to explain something to a non-technical person. Think of how you would explain HTML to your grandmother!

## On Site (Technical - Coding) Interview

**What to Expect:**

The interview will include topics such as coding, data structures, algorithms and computer science theory. You should know at least one language really well, preferably be C++ or Java (Python/C# are OK too). The interviewer will be interested in your knowledge of the following;

1. **General Coding Ability** - How well you know your chosen language, how quickly you code and how well your code works.

2. **Computer science principles -** Data structures / algorithms and how they can be used in your solutions. As well as studying computer science principles, be prepared to discuss algorithms in depth - how complex is an algorithm, how to optimise it.

3. **Analytical Skills** - Analysis of your own code (can you spot a bug, can you tell the interview what is the most optimal solution?) and analysis of the question (do you understand it, do you ask clarifying questions)

4. **Problem Solving Skills** - They want you to think out loud. Tell them what you are doing and what steps you are taking to solve the problem.

**Please review the following topics:**

Google Code style guides: Make sure to check out our Google code style guides (C++/Python). For Java we don't publish anything official.

**Algorithms and Data Structures:**

- Big-O notations also known as "the run time characteristic of an algorithm". You may want to refresh hash tables, heaps, binary trees, linked lists, depth-first search, recursion. For more information on Algorithms you can visit: TopCoder
- Sorting: Know how to sort. Don't do bubble-sort. You should know the details of at least one n*log(n) sorting algorithm, preferably two (say, quick sort and merge sort). Merge sort can be highly useful in situations where quick sort is impractical, so take a look at it.
- Hashtables: Arguably the single most important data structure. You absolutely should know how they work. Be able to implement one using only arrays in your favorite language, in about the space of one interview.
- Trees: Know about trees; basic tree construction, traversal and manipulation algorithms. Familiarize yourself with binary trees, n-ary trees, and trie-trees. Be familiar with at least one type of balanced binary tree, whether it's a red/black tree, a splay tree or an AVL tree, and know how it's implemented. Understand tree traversal

## On Site (Technical - Coding) Interview - Continued

- Mathematics: Some interviewers ask basic discrete math questions. This is more prevalent at Google than at other companies because counting problems, probability problems, and other Discrete Math 101 situations surrounds us. Spend some time before the interview refreshing your memory on (or teaching yourself) the essentials of combinatorics and probability. You should be familiar with n-choose-k problems and their ilk – the more the better.

- Operating Systems: Know about processes, threads and concurrency issues. Know about locks and mutexes and semaphores and monitors and how they work. Knowabout deadlock and livelock and how to avoid them. Know what resources a processes needs, and a thread needs, and how context switching works, and how it's initiated by the operating system and underlying hardware. Know a little about scheduling. The world is rapidly moving towards multi-core, so know the fundamentals of "modern" concurrency constructs.

### A few last tips:

- Talk through your thought process about the questions you are asked. In all of Google's interviews, our engineers are evaluating not only your technical abilities but also how you approach problems and how you try to solve them.

- Ask clarifying questions if you do not understand the problem or need more information. Many of the questions asked in Google interviews are deliberately underspecified because our engineers are looking to see how you engage the problem. In particular, they are looking to see which areas leap to your mind as the most important piece of the technological puzzle you've been presented.

- Think about ways to improve the solution you'll present. In many cases, the first answer that springs to mind isn't the most elegant solution and may need some refining. It's definitely worthwhile to talk about your initial thoughts to a question, but jumping immediately into presenting a brute force solution will be received less well than taking time to compose a more efficient solution. **Always try to find the most optimal solution.**

GOOD LUCK!