# Interviewing @ Google Zurich!

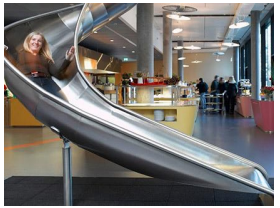
This guide is intended to add clarity to the role & responsibilities of a software engineer at Google. It will also give a clear overview of the recruitment process and help you best prepare for your upcoming interviews with Google. If you have any questions, please don't hesitate to get in touch.

THE OFFICE, PEOPLE AND PROJECTS

THE ROLES AND RESPONSIBILITIES

THE RECRUITMENT PROCESS

PREPARING FOR YOUR INTERVIEW
+ Technical Prep
+ Interview Tips



EXTRA PREP YOU MIGHT FIND HELPFUL

## The Office

Zurich is Google's largest engineering office in Europe, the Middle East and Africa. We like to think of ourselves as "the real Mountain View." Not because we're the company's engineering headquarters for EMEA, but because of the views out our top-floor Sky Lounge windows. Our office has a different design theme on each floor, massage stations, two gyms, micro-kitchens, a sleeping room and restaurant-quality food.

## The People

With Googlers coming from across the globe, Zooglers are an especially diverse bunch. Take a two-minute walk through the cafeteria, and you'll probably hear more languages than you can count on both hands. The very high level of expertise among our Zooglers is one of the most appreciated facts why Googlers like to work in Zurich. We are proud to have an environment where everyone can learn from each other. You can also find some cool videos with Zooglers on the left.

## The Projects

Googlers in Zurich do important work across our technology groups, and our launches have included Maps for EMEA, Search refresh, key Gmail features like the priority inbox and account abuse protection, and content ID and monetization products for YouTube. Other key projects we have in Zurich include Shopping, Ads, Calendar and Spam & Abuse

**Still want more info?**

Office location & Streetview

Office pictures

Zurich information

YouTube video

Life @ Google

Meet Sascha, Erik & Lina

## The Role

You can start at our Google Careers Page, which is broken down per location. It has real-time and detailed requirements for our needs across all of the projects. The basic requirements are summarized as, excellent coding skills (usually C++ and Java), with a strong foundation in computer science theory. This includes competencies in data structures & algorithms.

## The Responsibilities

Software Engineers at Google are responsible for the whole lifecycle of a project. Depending upon the project you join, you could be involved in research, design, planning, architecture, development, test, implementation and release phases.

**Ultimately you are going to help build Google products.**

## To Summarize

Google is and always will be an engineering company. We hire people with a broad set of technical skills who are ready to tackle some of technology's greatest challenges and make an impact on millions, if not billions, of users. At Google, engineers not only revolutionize search, they routinely work on massive scalability and storage solutions, large-scale applications and entirely new platforms for developers around the world. From AdWords to Chrome, Android to YouTube, Social to Local, Google engineers are changing the world one technological achievement after another.

**Still want more info?**

Google jobs

A Google engineers career

Ask a Google engineer

Meet other Googlers

Research @ Google

**Get a Googler's Perspective:**

*" My first few months at Google have all been about getting up to speed quickly. After the induction, I went straight into building a prototype for my project. I've been developing my code to the Google code styles and am now adding to the Google code base. I will be starting to use new technologies like Mapreduce and Flume and learning to quantify my performance and improvements. When I started the atmosphere was rich in learning opportunities, challenging, and rewarding.*

*With the support of the team it wasn't long before I was designing, creating and maintaining code to help support and contribute to the teams end goals. It's very evident that we are responsible for the full life cycle when you are a Software Engineer at Google. "*

## How we Hire

Google's interview process is the same globally. This consistent global approach allows engineers the opportunity to develop their future career within Google both geographically and across project changes. You can find a quick overview of the hiring steps in this guide. Your recruiter will conduct an initial call with you and be your partner during the whole process, keeping you updated throughout. Please bear in mind that things do take time.

## The Process

**Phone Interviews**
Up to 2 x 45 min interviews with a Google Engineer. A Google doc will be used as a virtual whiteboard for coding and algorithmic problems.

**On Site Interviews**
Up to 5 x 45 min interviews. You will use a whiteboard for discussions and coding tasks. You will also have lunch with an engineer (a great time to ask questions) and have a tour of the office.

**Committee Review**
An independent committee of Googlers review the Interview feedback. They ensure our hiring process is fair and that we're holding true to our "good for Google" standards as we grow.

**Offer Review**
We submit your resume, interview feedback, committee feedback, along with references, and transcripts to be reviewed by our senior management team.

**Offer**
If everything is approved, we will make you an offer, confirm it in writing and talk to you about possible relocation, visas and starting work at Google!

## Project Matching

As we assess you, your skills and strengths throughout the interview process we consider potential project matches here at Google. We take into consideration your interests as well as the current needs of the various Google projects before making a project match. It is not uncommon to match to several projects and sometimes we will arrange for a call with a Tech Lead of a project, so all parties involved can discuss further to help find a match.

**Still want more info?**

Interviewing @ Google

Hiring process for students

## Before you Start

Before you get started preparing for your interview, you can get an overview of interviewing with us, here. We highly recommend that you do your research about the interview process at Google, and a great place to start would be finding out how we hire. You can also check out Dean Jackson's ACM article on technical interviews at Google for great tips on the process and getting in some practice.

## Interview Questions

Interview topics may cover anything on your resume, especially if you have stated that you are an expert. Be prepared to write **code on a whiteboard**, build and develop complex algorithms, analyze their performance characteristics, data structures, logic problems and systems design. Finally, know your core computer science principles - hash tables, stacks, arrays, API's, Objected Oriented Design & Programming etc. Computer Science fundamentals are prerequisites for all engineering roles at Google, regardless of seniority, due to the complexities and global scale of the projects you would end up participating in.

## Where to Focus

Focus on demonstrating your problem solving skills, applied to the question asked. If it's a coding question, providing efficient prototype code in a short time frame to solve the problem is the key. If it's a design question, working with your interviewer to create a high-level system, at times perhaps diving deeper on particular salient issues, is what matters. If it's a general analysis question, show that you understand all particularities of the problem described and (where applicable) offer multiple solutions, discussing their relative merits. Every Interviewer ultimately wants to answer the question of whether they'd be comfortable working with you in their team.

**Still want more info?**

About us
The Google story
How we hire
Tech interviews @ Google
Google Developers

**Sign up for an online interview coaching session** with a Google engineer before your on site interview here!

# Technical Preparation

**Coding:** You should know at least one programming language really well, preferably C++ or Java. For specific projects, we do also use C and Python but these are normally secondary languages at Google. You will be expected to write code in most of your interviews. You will be expected to know a fair amount of detail about your favorite programming language. Make sure to check out our Google code style guides. You will be expected to know about API's, OOD/OOP, how to test your code, as well as come up with corner cases and edge cases for yours and other peoples code.

**Algorithms:** You will be expected to know the complexity of an algorithm and how you can improve/change it. Big-O notations also known as the run time characteristic of an algorithm. If you get a chance, try to study up on fancier algorithms, such as Dijkstra and A*. For more information on algorithms you can visit TopCoder.

**Sorting:** What common sorting functions are there? On what kind of input data are they efficient, when are they not? What does efficiency mean in these cases in terms of runtime and space used? E.g. in exceptional cases insertion-sort or radix-sort are much better than the generic QuickSort / MergeSort / HeapSort answers.

**Data structures:** You should study up on as many other structures and algorithms as possible. You should especially know about the most famous classes of NP-complete problems, such as traveling salesman and the knapsack problem. Be able to recognize them when an interviewer asks you in disguise. Find out what NP-complete means. You will also need to know about Trees, basic tree construction, traversal and manipulation algorithms, hash tables, stacks, arrays, linked lists, priority queues.

**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 elementary probability theory and combinatorics. You should be familiar with n-choose-k problems and their ilk – the more the better.

**Still want more info?**

Tech interviews @ Google

Google style guide

**Sign up for an online interview coaching session** with a Google engineer before your on site interview here!

# Technical Preparation

**Graphs:** To consider a problem as a graph is often a very good abstraction to apply, since well known graph algorithms for distance, search, connectivity, cycle-detection etc. will then yield a solution to the original problem. There are 3 basic ways to represent a graph in memory (objects and pointers, matrix, and adjacency list); familiarize yourself with each representation and its pros/cons. You should know the basic graph traversal algorithms, breadth-first search and depth-first search. Know their computational complexity, their tradeoffs and how to implement them in real code.

**Recursion:** Many coding problems involve thinking recursively and potentially coding a recursive solution. Prepare for recursion, which can sometimes be tricky if not approached properly. Practice some problems that can be solved iteratively, but a more elegant solution is recursion.

**Operating systems:** You should understand processes, threads, concurrency issues, locks, mutexes, semaphores, monitors and how they all work. Understand deadlock, livelock and how to avoid them. Know what resources a process needs and a thread needs. Understand how context switching works, 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.

**Front end engineers:** Refresh JavaScript skills, CSS, AJAX, HTML and GWT. Understand user-interfacing and techniques. Have a good sense of how the front and back interact and tie together. Cross Browser knowledge could also help.

**System design:** System design questions are used to assess a candidate's ability to combine knowledge, theory, experience and judgement toward solving a real-world engineering problem. Sample topics include: features sets, interfaces, class hierarchies, distributed systems, designing a system under certain constraints, simplicity, limitations, robustness and tradeoffs. You should also have an understanding of how the internet actually works and be familiar with the various pieces (routers, domain name servers, load balancers, firewalls, etc.). For information on system design, you can look here, Also understand How search works.

**Still want more info?**

Tech interviews @ Google

Scalable Web Architecture & Distributed systems

How search works

**Sign up for an online interview coaching session** with a Google engineer before your on site interview here!

# Interview Tips

### Substantiate
Make sure that you substantiate what your CV/resume says – for instance, if you list Java or Haskell as your key programming language, questions about this are fair game and may be asked of you.

### Explain
Explain your thought process and decision making throughout the interview. 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. We want to understand how you think. This would include ***explicitly stating and checking any assumptions*** you make in your problem solving to ensure they are reasonable.

### Clarify
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 you see as the most important piece of the technological puzzle you've been presented.

### Improve
Think about ways to improve the solution that you present. In many cases, the first solution that springs to mind isn't the most elegant and may need some refining. It's worthwhile to talk through your initial thoughts with the interviewer. Jumping immediately into presenting a brute force solution will be less well received than taking time to compose a more efficient solution.

### Practice
Make sure you practice writing code on paper or a whiteboard. Be sure to test your own code and ensure it's easily readable without bugs.

### Ask Questions
At the end of the interview, most interviewers will ask you if you have any questions about the company, work environment, their experience, etc. It's clever to have some pre-prepared for each interview.

**Get some practice!**

To practice for your interview, you may want to try:

Google Code Jam questions -- samples from Google coding competitions.

Take Dean's advice! Try practicing coding in a Google doc or on a whiteboard with a friend.

# Recommended by Googlers

## Books

Programming Interviews Exposed: Secrets to Landing Your Next Job
John Mongan, Eric Giguere, Noah Suojanen, Noah Kindler
John Wiley & Sons, 13.11.2012

Programming Pearls
Jon Bentley
Pearson Education, 01.09.2000

Introduction to Algorithms
Thomas H. Cormen
Mit Press, 2009

Cracking the Coding Interview
Gayle Laakmann McDowell
CareerCup, LLC, 2011

## Online resources

**Google Publications:**

The Google File System

Bigtable: A Distributed Storage System for Structured Data

MapReduce: Simplified Data Processing on Large Clusters

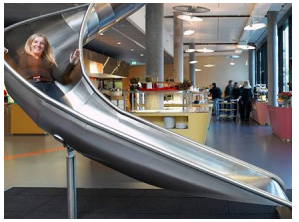
Google Spanner: Google's Globally-Distributed Database

Google Chubby

**Google Education**

Guide for Technical Development

**Sign up for an online interview coaching session** with a Google engineer before your on site interview here!



**Still want even *more* info?**

Google books

Google publications

Google scholar