How to Get a Software Engineer Job at Google and Other Top Tech Companies

medium.freecodecamp.org/how-to-get-a-software-engineer-job-at-google-and-other-top-tech-companies-efa235a33a6d

Hi everyone!

I've already talked about how I personally got a software engineer job at Google in a few videos on my programming-education YouTube channel called CS Dojo.

However, a lot of people still ask me about how to get a job at Google as a software engineer. So, here's my article explaining the strategy I would use.

You can use the same strategy to get a software engineering job not just at Google, but also at other top tech companies like Amazon, Microsoft, and Facebook.

In this article, I'm going to show you the 6 step process for landing a job at one of these companies. Along the way, I'm also going to discuss:

- How to learn to code in the first place
- What to learn after learning to code
- How to get your first coding job or internship
- The best ways to apply for software engineer jobs
- How best to prepare for programming interviews
- Is it important to get a computer science degree?
- Is it important to go to a top university?

Okay, let's get started!

Step 1: Learn to code

This is the absolute minimum qualification you need to become a software engineer.

For this, I'd recommend you get started with an interactive website such as Codecademy and freeCodeCamp. You can learn most of the programming fundamentals from these sites. I also heard that SoloLearn is good, too.

After that, I would use video tutorials to learn more advanced topics. I recommend websites like YouTube, Pluralsight, Lynda.com and Udemy for this. On these websites, you should be able to find tutorials on topics like:

- Web development
- Mobile development
- Game development

depending on your interest.

But wait, what programming language should I learn first?

My short answer would be, pick either JavaScript or Python, but it really depends on your interests. I have a longer answer to this question in this video.

Step 2: Work on a few personal projects

After you've followed a few programming tutorials, you should build a few personal projects to practice using what you've learned.

For this, you should find something you're interested in building.

For example, if you like photography, maybe you can build a portfolio site for all your photos. If you like trading stocks, maybe you can build a system that analyzes stock charts for you. Or, if you enjoy problem-solving, you can try competing in a coding competition.

When you work on a project, first build as much as possible by yourself. Then, if you get stuck, get help from others by using either online or offline resources. For example, you can use Stack Overflow to ask specific, technical questions if you get stuck on something programming-related.

Step 3: Get your first programming job or internship

Once you've built a few personal projects, you should be able to get your first programming job or internship. With it, you'll have some experience before you start interviewing with top tech companies.

It is possible to get your first job at one of the top tech companies, but it is far easier and more common to get it at a less well-known company first.

The best ways to apply for software engineer jobs

Simply applying for jobs online is not necessarily the best strategy, because that's what pretty much everyone else does.

Instead, I'd recommend using LinkedIn and in-person networking to get your first programming job.

On LinkedIn, first find recruiters of the company you're interested in working at. Then, ask them if you'd be qualified for the position you're interested in. You should also ask them how you can better prepare yourself if you're not qualified yet.

For in-person networking, I'd recommend websites like Meetup to meet engineers and recruiters at local companies.

Wait, wait, but that's not all.

Using LinkedIn and in-person networking work well if you're applying to small-to-medium-sized companies. However, I've found that these strategies are less effective for more popular companies such as Google and Facebook.

For these big companies, instead, I recommend combining the following three strategies:

1. Career fairs and recruiting events at universities near you.
2. Get referred by friends who work at one of these companies
3. Simply apply online.

Combining all of these strategies will help you increase the chance of getting an interview with one of these top tech companies.

By the way, I'm not exactly sure why LinkedIn is less effective with these big companies, but I think it might be because they get too many messages. Basically, they're too popular there :P

Step 4: Learn data structures and algorithms

Top tech companies like Google and Microsoft often ask questions about data structures and algorithms in their interviews. So, you should learn them if you haven't yet.

To learn the basics, I'd recommend my video series on data structures and algorithms. It's all on YouTube. I've made these videos to make it as easy as possible to get to know this topic.

Since there are only 7 videos in this series, you'll need more material to learn more advanced topics.

There are several popular options for this, including:

You should try a few of these and continue with the one you like the most.

Step 5: Prepare for coding interviews

Coding interviews at companies like Google and Microsoft are hard, but it is somewhat straight-forward to prepare for them.

Once you have a solid understanding of data structures and algorithms, I'd recommend these three resources for practice:

After practicing on your own for a few weeks, you should start doing mock interviews.

How to do mock interviews

Basically, pair up with your friends, and give each other problems from the resources I mentioned above.

Then, solve each problem on paper or on a Google doc. Explain your solution to your friend.

Make sure to practice both as the interviewee and interviewer so that you can see what it's like to be in the interviewer's shoes.

Once you do 20 or so mock interviews, you should be ready to start interviewing with the companies you want to work for.

Step 6: Apply, apply, and apply again :)

Use the three strategies I mentioned above to apply to top tech companies:

- recruiting events / career fairs
- getting your friends to refer you
- and applying online.

If you don’t get in the first time, don’t worry. In fact, you should expect some failures since getting into any of these companies is competitive.

For me personally, I needed to apply to Google 5 times before I landed my software engineer job there.

That’s it! Here’s a quick recap:

- Step 1: Learn to code
- Step 2: Work on a few personal projects
- Step 3: Get your first programming job or internship
- Step 4: Learn data structures and algorithms
- Step 5: Prepare for coding interviews
- Step 6: Apply, apply, and apply again :)

Wait, is it really that simple?

Yes, it is. Still, it takes a lot of time and dedication to go through all 6 steps.

A few more notes:

Do I need to get a computer science degree?

The short answer is no. However, getting a computer science degree helps a lot. If you take a solid CS program, it takes care of most of Step 1, 2, 3, and 4 (learning to code, doing personal projects, getting your first programming job or internship, and learning data structures and algorithms).

Without taking a CS degree, you’ll just need to learn more on your own.

Also, please note that even with a CS degree, it takes a lot of work to get a software engineer job at one of these companies.

Do I need to go to a top university like MIT, Stanford, Carnegie Mellon, etc.?

Again, the short answer is no. It probably helps a little bit, but it's far from necessary.

Laszlo Bock, a former Senior Vice President of People Operations at Google, also agrees.

According to the book he wrote, Work Rules!, Google prefers top-performing students at less-known schools over mediocre-performing students at top schools like MIT.

I think this makes sense, because if you're smart and dedicated, it shouldn't matter too much which school you went to.

Do I need to have a high GPA?

The short answer is no, again.

Having a high GPA probably helps a little bit for getting an interview as well, but it's probably much more important to have solid practical experience and interesting projects to show on your resume.

In fact, according to Work Rules!, Google used to care a lot about candidates' GPAs in the past. However, they've found that having a high GPA is not strongly correlated to having a strong workplace performance in the long term. After that finding, they stopped placing such a strong emphasis on GPAs.

What do I need then?

Basically, all you need is strong coding and problem-solving skills, a good understanding of CS fundamentals, and interesting projects and experience to show on your résumé.

You'll be able to build all of these if you follow the 6 steps I explained above.

Wait, how can I write a goodrésumé **though?**

If you’d like, you can use the resume I used to apply to Google as a template.

That’s the actual résumé I used to get a job at Google as a software engineer. It should really have been one page, so I would say my résumé was too long. Still, feel free to use it as a starting point.

Additional resources for writing a résumé**:**

- There’s a nice overview about this topic on CareerCup here.
- My friend Zhia Hwa Chong, who is a software engineer at Twitter, has a nice article about it here.

Okay, good luck, and thank you for reading this article!