A CC Student in the Land of Tech: Navigating Interviews

Written by Omer Baror, class of 2006
For questions about this document, contact the Career Center at CareerCenter@coloradocollege.edu.

Objective – why are you writing this?

The objective of this document is to help you, a current CC student, navigate the very likely confusing and overwhelming process of interviewing for a job in technology.

It was at least pretty confusing to me, when I was a CC student! I didn't know whether or how to find internships, where to apply, or how to get through the process. CC is a wonderful place in a lot of ways, but one advantage that students at larger universities have is technology companies (like Google, Amazon, Meta, Apple, etc.) coming to *them* as part of career fairs and other events, so it can seem like a less anonymous and amorphous process for them.

I strongly believe that the tech industry needs more liberal arts alumni in its ranks, because a lot of the more critical skills to make technology both successful and good for the world are liberal arts skills.

Okay, but Omer, who are you?

Hi. I'm Omer. I graduated from CC in 2006, not with a degree in computer science but one in comparative literature. I got a master's at UCLA in computer science after that then went off to work in the industry. Early on, I had a hard time finding my footing (I applied to Google three times before I got in), but I've since had more success in my career.

I currently am fortunate to work at Waymo, which builds autonomous vehicles, one of the most interesting and socially impactful problems in the world. At Waymo, I'm a senior director of engineering and the lead of the Planner organization. You could think of the software in an autonomous vehicle as having two main pieces: what you see around how and how you choose to act based on what you see. Planner is the "how you choose to act" part.

So, how does the interview process work?

One caveat: the experience I'm drawing from is mostly interviewing for software engineering jobs in several companies and being an interviewer and on hiring committees for Alphabet (Google and Waymo). I've been an interviewer for a variety of roles (product manager, program

manager, administrative business partner, HR business partner, and more), but the vast majority of my experience is software engineering. Depending on the company and role you interview for, you may find the process different. To learn more about these processes, a good idea is to connect with other alumni as well as recruiters through events (including me: see contact info at the bottom). You should also contact the Career Center to help make these connections.

Step 1: applying

Job opportunities can be found on multiple different platforms, including Handshake and company websites. Most technology companies have careers pages, including both full time and internship roles. Here's Google's. Here's Apple's. Here's Waymo's. In fact if you go online and type in <your-favorite-company>.com/careers , you'll probably find a career's page. These pages are overwhelming. There are **so many** companies and **so many** jobs in them!

Your first step is to determine whether you are eligible for internships or full-time roles. Usually, internships are intended for current students, and full-time roles are for individuals graduating within a year, so filter on jobs pages by which one you're looking for. Internships are a great stepping stone to a full-time position, either in the same company or in a similar role at another company.

The second step is to think about what role you might want. This sounds like a permanent decision, but it's really not. I know many people who have moved from one role to another, even a very different one.

Here are the most common roles I've seen in my day-to-day work (but of course there are many others):

- **Software engineer** (i.e., the "how") – someone who solves problems directly by writing software. This is the job you want if you want the closest feeling of being able to point to something and say, "I built that!" You need to know how to write code, of course (the exact language(s) depends on the company). You also need to know (a) how to analyze and solve problems and (b) how to communicate and collaborate with those around you.
- **Product manager** (i.e., the "what") – someone who cares deeply about what the actual product is: will it satisfy a problem that people have? How will they use it? What problems will they encounter when using it? How will it make money for the company? This is the sort of role you want if the thought of running your own company one day is exciting (even if still terrifying) for you. You need to understand technology, but many roles don't require much writing of software. But you should have a very strong ability to analyze abstract problems and communicate and collaborate. You should also have a gut sense of people's relationship to products, e.g., how you might improve a product you use so that more people would want to use it.
- **Program manager** (i.e., the "when") – someone who makes sure that the team runs as effectively and efficiently as possible. You often don't build things directly, but you enable everyone else to be successful. A good program manager is often a force multiplier in an organization: organizing a team of five well enough that they can do

more than what a disorganized team of ten would do. This is a good role for you if you are meticulously organized and love helping the people around you succeed. Some program manager roles do require some amount of software engineering capability (typically called "technical program managers"). Mostly the skills you'll need are around keeping yourself and others organized to make sure things happen on time and that nothing slips through the cracks.

If any of those sounds like exactly what you want, look for roles at companies that sound interesting and apply.

If more than one sounds interesting, or none of them sounds perfect, try applying for a few of them. It may feel weird to apply to a job that you're not sure you want, but: a job interview is an opportunity both for the company to interview you and for you to interview the company. You'll learn a lot from the sorts of questions that the interviewers ask, and good interviews usually leave you time to ask questions too, so you can learn more about the job.

Once you've decided on roles, it's time to consider some companies and apply. There are a lot of companies, from tiny startups to very large companies. It's usually a good idea to apply to several roles, across multiple companies (see "Apply to Jobs Strategically" below for some tips). It will give you good experience and maximize the likelihood that you find a role and a company you're excited about.

Step 2: recruiting pass

Any reasonably large company will have a recruiting team who look at applications and determine: (a) whether you're possibly a fit for the company and (b) where specifically you might be a fit. They have a good sense of what open roles a company has, which is important because careers pages can be overwhelming, out of date, or just have general role postings not ones for specific teams.

In many cases, the recruiter will reach out to you directly to talk and learn more about you, your skills, and your interests. They'll also be trying to sell you on the company, so you're excited if you do get an offer.

It's worth noting that many recruiters are not deeply familiar with technology. That means, when you talk to them (and in your résumé), you should focus on key attributes that the role needs and that you have (e.g., I can program in Python, or I did a personal project using ML).

Be kind to your recruiters. They're human beings (who are treated poorly surprisingly often). And honestly they have a lot of sway in how your application process goes. Of course, working with a recruiter will not automatically lead to an offer. Connecting with recruiters can be a valuable way to learn about a company and available opportunities, and they can support you throughout the application process, but they cannot guarantee you a position.

Step 3: phone screen

If the recruiter identifies a potential fit between you and one or more teams at the company, they'll often set you up with a single interview to assess whether it's worth doing a full set of interviews. For smaller companies, this is often (but not always) with a lead on a team you could be working on. If this interview doesn't go well, the company will likely say "no." If this interview does go well, you'll likely move on to the next phase. If it's in between, you may be asked to do a second phone screen.

Treat this as a typical interview (see next phase).

Some companies have also augmented or replaced this phase with a take-home project, where you try to finish a project within some allotted time.

Step 4: full interviews

If you pass the phone screen, you'll often be asked to interview with several people (often 4-5) on the same day or two. Pre-COVID, companies would fly you out to interview in person, but now some companies just do several video call interviews on the same day or two.

Note: for internships, companies may forgo this step or just do a couple of phone screens.

Interviews depend heavily on the role. I'll cover software engineering here, since that's where almost all of my experience is. Luckily, for basically any role, you'll find amazing resources online, e.g., Software Engineer, Product Manager, Program Manager.

It's important to note that you don't have to do perfectly on all of your interviews. First, that's because you're doing several interviews, so even good candidates will mess up at some point. Second, interviewers want to see you working through problems, so struggling with a problem (and then figuring out how to overcome it) is actually a great sign of success. Regardless of role, it's very important to talk through your thought process: how are you thinking about what the interviewer just said? Do you have any follow-up questions? Do you have an idea even if there's probably a better answer?

Just for software engineers: if you're not interested in a software engineering role, feel free to skip to the next phase.

The most common type of software engineering interview is the coding interview. In a coding interview, the interviewer gives you a toy problem (i.e., one that's possible to solve all or part of during an interview), and then your job is to solve it. You're often given some place to write code, but it isn't your normal text editor or IDE. These interviews can be pretty intense until you get the hang of them. Here are a few things that have helped me:

- **Choose the right programming language** – ask about using the programming language you're most comfortable in, not just what the job will mostly entail. Some interviewers will want to see you write in a specific language, but most in my experience

won't. As an interviewer at Google, I let people interview in whatever language they want. I wanted to see if candidates could solve problems and write good code in any language. If so, I was confident we could teach them whatever programming language we needed them to know.

- **Ask follow-up questions** – many coding interviews start out being under-specified, because in practice, most projects you take on are initially vague too, and the interviewer wants to see if you'll get stuck.
- **Think out loud** – this is a hard skill to master if you don't do it naturally, but it's critical. As an interviewer, if I'm sitting in silence while you think, I have no way of evaluating you, and I also have no way of helping you if you're going down the wrong path.
- **Practice over and over again** – being good at a coding interview is correlated with being a good software engineer, but they're different skills, so even if you're good at writing software, you'll benefit from practicing interviewing too. There are a lot of resources online with examples of coding interviews. One thing that works well is finding someone else to interview you (e.g., a friend who is also applying for software engineering jobs).

Step 5: company decision

Once you've interviewed, the company will use some process to reach a decision. For example, the hiring manager may look at your and other packets and decide, or there may be a committee of people who review your packet and come to a decision together. A good recruiter will keep you apprised of where you are in this process.

Step 6: offer

If the decision is to hire you, then the company will put together an offer letter for you (often the recruiter will coordinate this process). The offer will include lots of details, though usually you'll care most about:

- The specific role (especially for a bigger company, where you may not have known the exact role when you applied and interviewed)
- The financials
    - Your salary – how much money you'll make before taxes in a year
    - Annual bonus target – most but not all companies give annual bonuses to employees in good standing. The exact amount depends on many factors, including how well the company is doing and your own performance. This is usually just an estimate of what you'd get if you performed well (not badly but also not amazingly)
    - Sign-on bonus – sometimes a company will give you some extra money to accept their offer as a way of enticing you more
    - Equity – how much of the company's stock you'll get (if any). Equity typically vests over 4 years, so the number may seem big, but you'll only get it in smaller pieces over a long time. Unvested equity is equity that you can't do anything with

and that you'll lose if you leave the company. Vested equity is equity that you can sell (if the company has gone public) and that you keep even if you leave the company. Like annual bonuses, in many companies, most or all employees in good standing will get an equity refresh every year (that also vests over four years). The amount depends on several factors: your performance, your potential, and how well the company is doing.

Many companies will be willing to negotiate some or all parts of an offer, especially if you're an exciting enough candidate (e.g., one who has a strong resume or did very well on interviews). It often doesn't hurt to ask even if the answer is ultimately no.

Personally I think your role is way more important than the financials. I've often passed up more money to work on what I am most passionate about, because I've considered my time to be the much more valuable resource that I don't want to squander. I've also found that, if you are successful in your role, your initial offer is a small fraction of the overall money you'll make in a given company.

The CC Career Center can help you navigate the process of evaluating and accepting an offer.

Step 7: your decision

Okay, you have an offer, now you have to decide. Do you want to work here? If you have a few offers, this can be a hard choice. You're trading off options with only partial information. For me, the biggest considerations were: (a) where I think I can learn the most and (b) where I think I can have the most impact on the world. When I was younger, (a) was more important to me, because I figured I'd need it in order to do (b). Now I look for a mixture of (a) and (b). In the future, I may focus just on (b).

But, you may have a totally different set of criteria. Or you may just go with your gut. Ultimately, this is a very personal decision, so nobody can tell you what to do other than yourself.

How should I prepare?

Great candidates usually start well before the application to prepare. Here are a few things I recommend.

Create software projects for fun

One of the most important things I did before landing a job in the industry is fell in love with software engineering. I took on several side projects, often with friends (including while at CC!). **We didn't pressure ourselves to build anything amazing or thought we were building the next Google or anything. The goal was to have fun and to learn.** Through personal projects, I learned several languages (C/C++, Python, Perl, Ruby, Javascript, …), all parts of the web stack (database, backend, frontend, user interface), and generally how to

solve problems. I added a few of the projects I was most proud of (and a link to the code) to my resume. Ultimately, the projects helped my application stand out, helped me interview, and most importantly helped me be a good software engineer once I got a job.

While this experience taught me a lot about software engineering, we also needed to do a lot of product and program management, and even some user experience design. So no matter what role you're looking for or thinking about, partnering with some friends on projects is a great way to get relevant experience.

Practice interviewing

Interviewing is meant to correlate with being good at your job, but it's a skill in its own right, and just like any skill, most people start out being bad at it. You can find lots of example questions online. For some examples, see the links in the Phase 4: full interviews section.

One thing to watch out for is low quality online resources. At least for applying for Google, good interview question lists typically avoid very general questions ("what is your worst quality") and "gotcha" questions (that require guessing what's in the interviewer's head, e.g., "why are manhole covers round?"). So if your online resources have a lot of questions like these, keep looking. Obviously, I can't guarantee that you won't be asked these questions, but in my experience, questions are much more specific and targeted to your role.

The CC Career Center can help you practice, and they also offer resources, like Big Interview, to help you prepare.

Apply to jobs strategically

Consider the order in which you apply for jobs to maximize the likelihood that you'll land the job you're excited about and also avoid hard choices on the path there (e.g., a company you're less excited about has given you an offer, but that offer expires before you interview with a company you're more excited about).

As one example, when my wife was applying for new software engineering positions, she split the companies into three tiers:

- Tier 1: least excited but would be willing to work there
- Tier 2: pretty excited but not the dream job
- Tier 3: dream job

She first applied to tier 1 companies. Around the time she started interviewing with tier 1 companies, she applied to tier 2. When she was hearing from tier 1 companies and was starting to interview with tier 2 companies, she applied to tier 3 companies. Following this strategy had two nice benefits:

a) She got a lot of experience practicing interviewing on tier 1 companies, so by the time she applied to tier 3 companies, she had hit her stride at interviewing.

b) By the time tier 1 companies' offers were expiring, she was already getting tier 2 offers, and same between tier 2 offers expiring and getting tier 3 offers. So, she never had to make the hard choice of taking a job at a company she was less excited about just because she didn't know the outcome yet of a company she was more excited about.

Ultimately she got and accepted an offer from one of her tier 3 companies (Google).

This exact strategy may or may not work for you. It worked for her because she was willing to dedicate the time to apply to many companies, and she had an appealing enough resume and interview skills that she could expect to get multiple offers. You may be in a different position, or looking for something different, so a different strategy may be better for you. **The key is to think through your strategy up front rather than going in blind (e.g., applying to the company you're most excited about first just because you're so excited about them).**

Any other tips?

- **Don't let rejections stop you:** I applied to Google three times before I got in. I knew I really wanted to work there, so I kept trying. I won't promise I was always chipper about hearing "no" or that I never doubted myself for a second, but I kept trying every couple of years, and ultimately made it in.
- **It's okay to feel imposter syndrome:** to be honest, I still feel imposter syndrome on a very regular basis. Many people that I know feel it too. My best personal strategy for dealing with imposter syndrome is, whenever I feel it, acknowledge that's what I'm feeling, and then force myself to do *something* productive: what is the very next step I should take? In an interview, that may be asking a clarifying question or saying the first thing on your mind, acknowledging that it's probably not perfect.
- **Come prepared with questions to the interviewer:** this is an opportunity for you to learn more about the company and the role. And as an interviewer, if a candidate came with good questions, I took it as a sign that they were engaged and excited about the role.
- **Take something from the experience:** interviewing is definitely stressful. It's also an opportunity to meet intelligent and accomplished people to discuss intellectually interesting problems. My best interview by far when I interviewed at Google was with someone who loved Python as much as I did. So, as stressful as it is, as much as it feels like so much of your future hinges on just these 45 minutes going well (it really doesn't), try to also treat this experience as an opportunity to learn about yourself, gain new skills, and meet some really cool people you may spend many years working with.

I have some more questions. What should I do?

I highly recommend three resources:

a) **Schedule some time at the Career Center.** It's a resource I didn't take enough advantage of when I was at CC.
b) **Talk to one of your professors.** Many professors have interviewed before and/or worked with students who have interviewed.
c) **Reach out to someone you know who is already in the industry.** In fact, you now know me! Please feel free to reach out with any questions. As a reminder, my name is Omer Baror. You can reach me at omer.baror@gmail.com or on LinkedIn.