

Preparation Cheat Sheet

Practice a ton of problems: code, code, code

"I don't care how smart you are; there is simply no substitute for practicing a ton of problems." - Professor Philip Guo

1. Any time spent coding is good interview prep time
2. Implement common **data structures**, **algorithms**, etc.
3. Code in **one** chosen interview language

Code at a Whiteboard

1. **At school?** Practice somewhere on campus
2. **At office?** Find a conference room and use it after-hours
3. **At home?** There's even [sticky whiteboard paper you can get](#)
 - a. Group member Asim [using his to solve a bit manipulation problem](#)

No whiteboard?

Use pen and paper.

Phone interview?

Practice at your interviewer's location of choice. If they'll be using Etherpad, code up a couple of sample problems on Etherpad to get comfortable.

Practice writing real code

Write real code while preparing (*not pseudo-code*) and once you're finished—test your code and check that it compiles.

Work to get comfortable writing code that compiles and works on the first try.

Positive attitude

There will be jobs you don't get, use them as a learning experience.

Getting stumped during an interview is a great signal you should drill hard on that specific topic.

Do more real interviews

A few months ago, I offered a piece of advice to a group member just starting his job search. When you are interviewing for jobs—especially if this is your first time code interviewing—don't do your first interview with your #1 dream company. Quite frankly:

The more coding interviews you've completed recently, the better you'll do at your next coding interview.

Forget number of hours slept the night before, which code interview book you bought, or what algorithms you memorized. I'm convinced that the best predictor of code interview success is number of tough code interviews completed prior to your next one.

A Coding for Interviews member who had just gotten his first full-time gig out of school:

"Doing a bunch of technical interviews both on phone and in person before going to [new job] really helped me be confident and able to answer all their questions so well."

It's obvious how to capitalize on this effect. When you're looking for jobs, it's a sub-optimal strategy to run down your list from dream-job to okay :(job.

Reverse it. Interview as many times as is reasonable before your target job interview. In the worst case, you have a string of rejections and end up with a bunch of code interview experience. In the best case, you might learn you're actually compatible with a company you didn't thought you were before.

Of course, don't go overboard. There are plenty of companies out there, so you should be able to find enough interviews with companies you're interested in. 5-15 in-person interviews in a job search sounds about right. 25+ might be a bit extreme. That's a full time job!

More Preparation Tips and Tricks

1. [Preparing for a software engineering interview](#) by veteran ex-Googler Niniane Wang
2. [Coding Interview Tips](#) from Parker Phinney of InterviewCake
3. [How to prepare for technical interviews](#) by Dan Blumenthal, Director of Engineering at TripAdvisor
4. [How to be a star](#), great essay on long-term personal software engineering development by Dan Blumenthal
5. [What recruiters look for in a resume](#), also a good essay on long-term development from the perspective of when a recruiter looks at your resume years down the line