QuickStart Guides®

Navigating the Job Market:

- ✓ INTERVIEW PREP
- ✓ CODING INTERVIEWS
- ✓ AND ANALYZING JOB POSTINGS

Navigating the Job Market: Interview Prep, Coding Interviews, and Job Post Analysis

INTRODUCTION

Learning how to prepare for job interviews is a skill in its own right. Many competent software developers and programmers struggle with being interviewed, and it's no mystery why. Job interviews can be stressful, and when there is a technical element involved it can make them even more so. We have compiled some useful resources and advice to help you master coding interviews and analyze job postings to identify the best opportunities for your career.

INTERVIEW PREPARATION

If you really want to ace your coding job interview, you will need to do a bit of preparation. It is impossible to prepare for every eventuality, so don't wear yourself out trying to learn every concept under the sun. Instead, keep in mind these practical tips to help you get ready.

RESEARCH THE COMPANY AND ITS CULTURE

Part of your preparation must involve a fair bit of research into the company at which you want to apply. You should try to familiarize yourself with its mission, values, and any recent news. Try to gain insights into the culture of the company, and try to find out how they expect their staff to conduct themselves.

Find out about the products the company creates and the tech stack they use. There are websites such as Glassdoor that have actual testimonies from present and past employees, which can give you insights into the company's culture. Remember to take these reviews with a grain of salt though, as they are highly subjective.

Understanding the company's culture will help you tailor your answers and demonstrate why you would be a good fit.

EXAMINE THE JOB REQUIREMENTS AND RESPONSIBILITIES

This is probably the most important thing you need to familiarize yourself with before you even apply for the job. You must read the job description thoroughly and make a list of the key skills that you will need to exhibit, as well as the responsibilities that you would be expected to take on.

BRUSH UP ON YOUR TECHNICAL KNOWLEDGE

In the early stages of a job interview you will probably be expected to talk about the technologies that you currently work with. Sometimes it's difficult to explain concepts that we are familiar with, so it's a good idea to brush up on the basics. This means that you should think about reviewing programming languages, frameworks, and tools mentioned in the job posting. Make sure you have a solid understanding of the fundamentals and any specific technologies the company uses, so that you can answer any questions that come up.

PRACTICE COMMON INTERVIEW QUESTIONS AND BEHAVIORAL SCENARIOS

It is always a good idea to familiarize yourself with common programming interview questions and to prepare answers that showcase your problem-solving abilities and experience. The technical nature of a coding interview means that you will never be able to practice for every question, but the more you practice, the better your performance will be at the interview. Also consider practicing answering behavioral questions using the STAR (Situation, Task, Action, Result) method.

PREPARE A LIST OF QUESTIONS TO ASK THE INTERVIEWER

Many job hunters neglect this step when going for an interview. You should always have some questions ready for your interviewer, as it shows that you have a genuine interest in the role. Prepare a few thoughtful questions about the company, team, and role. This demonstrates your interest in the company, and it also shows that you are taking some initiative and you want more information to make a considered decision.

MASTER THE CODING INTERVIEW

There is a lot to cover in the coding interview. The level of technical knowledge that you need to showcase will vary from role to role, so you must be confident in your abilities related to the job posting that you apply for.

WHAT ARE CODING INTERVIEWS FOR?

The main reason for a coding interview is to assess your coding skills and problem-solving abilities. Your interviewers will also try to gauge how well you communicate and what your demeanor is like while you work. The bottom line is that you must write clean, efficient code. We will now go into detail and cover important aspects of your coding interview and some of the challenges that you need to be aware of.

Types of problems

Coding interviews generally involve problems that rely on your algorithmic and data structure skills. You will need to write code that effectively solves

problems, like sorting algorithms, graph and tree traversals, and string manipulation. Depending on the role that you apply for, you could also be expected to work with databases, networking, and operating systems.

Whiteboard coding
If you have never written out any code by hand before, now is a good time to start practicing! Technical interviewers sometimes have candidates write out a solution by hand and explain each step as they go. This is especially difficult if you have grown overly dependent on auto-correct and code completion in your IDE; figuring out a solution, writing it out by hand, and explaining yourself clearly may test your limits. Spend time ahead of your interview practicing writing code by hand and explaining it out loud.

Online coding
Sometimes you will need to do your technical interview remotely. To do this, you will need to use either an online editor or a modern IDE with screen sharing or Visual Studio's Live Share feature. There are other tools that you might need to use as well, such as HackerRank's CodePair, CoderPad, or CodeSignal.

Time limits
Coding interviews usually last around an hour, and there is normally more than one problem to solve in that time frame. A good way to practice is to set yourself a time limit when you are running through practice questions before your interview. This kind of training helps you to get more efficient with your time and more deliberate with your approach.

Follow-up questions
If the interviewer has questions about how you approached your problem, you can expect to spend a little bit of time discussing the finer points of the coding. This is where they will want to find out how your thought process works and what decisions you made along the way.

Once you understand what coding interviews are for, you will get a much better idea of what the interviewer wants to see from you. The key takeaway is that you must practice as much as you can ahead of your interview. You won't be able to predict what you are going to be tested on, but the more you practice the methods outlined above, the better prepared you will be.

KEY PROBLEM-SOLVING TECHNIQUES FOR CODING INTERVIEWS

When you are being interviewed for a programming role, it is important to be familiar with problem-solving techniques that will speed up your time to solution. Below are some brief explanations of problem-solving techniques that you should be comfortable with, or at least know of.

Divide and Conquer
This technique involves breaking up a problem into multiple subproblems that are easier to manage individually. Once you have all the smaller parts working properly, you can start linking them together and running the main solution.

Dynamic Programming
This is quite similar to the Divide and Conquer method, but it uses a table to help you keep track of your progress and to store your results, which you can then reuse. This is ideal for working on optimization problems. Popular examples that use this method include the Fibonacci sequence, the knapsack problem, and the longest common subsequence.

Greedy Algorithms
These algorithms are designed with optimal selection in mind. To work, they select the most efficient steps that provide the greatest benefit in the short term for the easiest or shortest path to the desired solution. Some well-known examples of this are Kruskal's and Prim's minimum spanning tree algorithms, Dijkstra's shortest path algorithm, and the Huffman coding algorithm.

Backtracking
Backtracking involves code that builds up a solution piece by piece and then reverses, or backtracks, when it reaches a dead end or an error. This is sometimes used with depth-first search solutions where combinatorial problems need to be solved. You can search for the eight queens problem, the traveling salesman problem, or Sudoku puzzle solving if you want to learn more about this technique.

There are many more techniques that you can learn to solve common programming problems, such as graph algorithms (which are used for solving problems with nodes and edges) and bit manipulation (which is used for operations like shifting, toggling, and masking). If any of these techniques are new to you, I recommend you look into how they work and where you are most likely to encounter them.

CODING INTERVIEW EXTRAS

In order to help you really nail your coding interview, I've gathered a few more tips that will help you be as ready as possible on the day.

Language Requirements
Remember that not all companies will have a specific programming language that they want you to use in your coding interview. Some might allow you to use your preferred languages to see how you best complete the tasks they give you. Other companies might have very specific requirements about what programming language you are allowed to use. So you need to have a clear understanding of what is expected from you.

Company Coding Challenges
Some companies are well-known for the types of coding questions they ask, so try to familiarize yourself with some examples if you can. Google and Facebook, for instance, ask very specific coding questions, and you can find examples on sites like Glassdoor and LeetCode where people that have actually done the coding interviews give valuable advice about what to expect.

Practice with real-world examples
It is a good idea to get as much practice as you can with generic coding questions, but as you approach the deadline for your interview, try out some real-world interview examples. LeetCode, CodeSignal, and HackerRank all have examples of actual questions that have been asked in interviews and are great resources for you to practice with.

ANALYZE PROGRAMMING JOB POSTINGS

In order to find the right job for your skill set, you need to understand exactly which roles you are applying for. This sounds easy enough, but there are often intricacies hidden in a job opportunity that you should be aware of so that you can acknowledge it in your interview. Nobody likes being caught off-guard, especially in an interview, so preparation is key.

IDENTIFY KEY REQUIREMENTS AND DESIRED QUALIFICATIONS

When you examine job postings, always start with the minimum requirements of the job. Look for the minimum qualifications and certifications that are required, and find out which essential skills are needed for the job. This information will help you create a custom application for the role, as well as prepare for the interview.

EVALUATE COMPANY CULTURE AND FIT

Not much thought goes into this when people are eager to start at a new company, but culture is important. Think about what your values are and whether the company aligns with them. For example, does the company work in an industry that you are not comfortable associating with, such as tobacco or alcohol? These are things that you should consider before applying.

ASSESS POTENTIAL GROWTH OPPORTUNITIES AND CAREER PATHS

Nobody wants to work in a position that has no scope for development and growth. If you are in a technical role, then there should be some upskilling and development prospects that come with the position. It is a good idea to find out about these elements of the job and how an employee can advance in the company, if at all.

CRAFT A TAILORED APPLICATION

To get your application in the front of the queue, customize it for the role you are applying for. If you have the skills they are looking for, then you need to highlight it in your application.

CUSTOMIZE YOUR RESUME AND COVER LETTER BASED ON JOB REQUIREMENTS

It is important for you to identify the skills mentioned in the job posting and adapt your application to underline your relevant programming skills, work experience, and achievements that match the job description. Many job applications are screened with automated systems, so the more keywords you match, the better your chances of getting called in.

DEMONSTRATE RELEVANT SKILLS AND ACHIEVEMENTS

When it comes time to show off your achievements, don't be shy. Prospective employers want to know what you have accomplished in your career, so make sure you have some experiences, skills, and achievements that you can share with them. Extra points if they are related to the role that you're applying for.

SHOWCASE YOUR PERSONAL PROJECTS OR CONTRIBUTIONS TO OPEN-SOURCE PROJECTS

Sometimes your passion for the work you do will shine through in your personal projects. If you have a portfolio or a pet project to share, make sure to select some areas of interest to show off. Highlight areas that display your programming skills and knowledge. And don't be afraid to talk about any collaborations or contributions that you have made to projects in your spare time.

ADDRESS POTENTIAL GAPS OR WEAKNESSES IN YOUR APPLICATION

You never want to be defensive if there are a few areas in your skills catalog that could do with improvement. Instead, openly acknowledge where you think you could improve, and talk about how you currently deal with those issues when they come up. The key is to be proactive and open about any concerns they might have, and be ready with some solutions.

FOLLOW UP AFTER INTERVIEWS

The hard work doesn't stop at the interview stage. If you really want your application and interview to shine, then you need to follow them up with a few things that will, hopefully, seal the deal.

SEND THANK-YOU EMAILS TO INTERVIEWERS

Sometimes it is appropriate to send a thank-you that expresses your gratitude

for the opportunity to interview. You will want to reiterate your interest in the role and say that you are looking forward to finding out the outcome. You will often leave a good impression and stand out from other candidates by being courteous and thankful. This is helpful even when you think the interview didn't go well, because our impressions are sometimes skewed by our nerves and self-doubt in stressful situations.

SEEK FEEDBACK AND USE IT FOR IMPROVEMENT

Unfortunately, you won't always get more communication from the company after an interview. However, if you feel that it's appropriate, you can sometimes ask for feedback after an interview. The worst thing that can happen is that they won't reply. If you do hear back from your interviewers after requesting feedback, you may get some helpful insights into your performance and what you can do to improve it.

MAINTAIN A PROFESSIONAL ONLINE PRESENCE

If you have an online presence for your professional life, then you will want to make sure it is up to date and presentable for potential employers. Keep your personal projects up to date and your profiles current with your latest certifications and achievements. This gives you the best chance if recruiters or HR staff are browsing your profile.

CONCLUSION

Coding interviews can be a very stressful process if you don't prepare yourself and practice with as many examples as you can. There are many ways you can prepare for a technical interview, and we hope the methods and tips that we have outlined will be helpful to you in your quest to land that dream programming job.

Be sure to thoroughly research the role you're pursuing, and practice interviewing until you feel you are ready for the real thing. If you have practiced and prepared yourself enough, then your chances for success are that much greater. Good luck!