

Five Interview Phases to find the Best Programmers

by Gayle Laakmann McDowell



Want to know the perfect interview questions for programmers? We do, too, but unfortunately there is no such thing as ‘the perfect’ set of interview questions.

The questions you ask will vary based on your organization, the need, and the role.

There may not be a perfect set of interview questions to find A+ programmers, but there is a basic flow you can use to maximize the efficiency of your screening process and end up with top tier programming talent.

I’ve approached this as a bottom-up approach, where you gradually screen out unqualified candidates quickly and easily, and then end up with a small set of great candidates. There are five phases in this bottom-up approach and ultimately the hiring manager can decide what phase they want to begin.

Phase 1 Interview Yourself

It may seem silly, but a lot of organizations skip this crucial pre-phase. Before you start interviewing people, you need to interview yourself about the role and it's needs. You need to ask yourself:

- What skills do you want?
- Why do you want those skills? Are they truly required?
- What will it take to get that skillset?
- Can you do without some of these skills?

Many companies ask the first question, but don't sufficiently explore the second, third, or fourth. Once you have figured out phase one, then it's time to move onto the next four phases of the interview process and pretty soon you'll have hiring great programmers down to a science!



Phase 2 Knowledge Validation

The Knowledge Validation Phase simply verifies that a candidate knows what they claim to know. It's asked under the assumption that nearly anyone who had worked in a particular area for the length of time this candidate has, would have learned this information.

Where To Use It

These questions are best used during the introductory call in the hiring process. The objective is simple: is this person's resume represented accurately? If someone can't pass these basic litmus test questions about their own defined skill set then this is likely not a candidate you want to pursue further.

Examples

- Your resume states that you've been doing search algorithms for a while now, can you tell me how a binary search tree works?
- I see you have five years writing in Java. Given that, what does "finally" accomplish in Java code?
- Based on your four years of Python scripting, can you tell me the difference between range and xrange?

Where It Goes Wrong

When you only ask one question about their resume experience and don't give them an opportunity to explain. If the candidate answers how you would expect, great. If they don't, give them a chance to follow up and explain why and their answer may completely justify their previous answer (or lack thereof).



Phase 3

Coding & Problem Solving Skills

The questions in this phase won't be horribly difficult but they'll give you some insight into a candidate's coding and problem solving aptitude. They won't help you distinguish whether a candidate is qualified or exceptional, but it helps you move the ball forward.

Where To Use It

These questions usually take place during a longer interview and offer the best insight into a candidate when paired with real-time coding challenges. An important note about this phase (and subsequent phases), is that every role is different, and for entry level roles, passing just phase one and phase two may be enough to qualify someone for a position.

Examples

- Write code to rotate a matrix by 90 degrees.
- Given a number and an array of integers, write code to check if the number is the mean (average), median (middle), or mode (most common number). It could be all of these or none of them.

Where It Goes Wrong

The big flaw here is that some of these coding questions may be so generic that the candidate has memorized them without actually understanding the workflow. That's why it is always best to ask a few of these basic questions to make sure they can actually assess the problem and write code to solve it.

This phase can also go wrong when the hiring team puts too much or too little value on it. For example, if you're seeking an entry-level position, then passing this phase could be enough. Or, on the flip side, if you're looking for a highly specialized position, then errors in this phase should be a strong indicator that the candidate isn't ready. Thus, be sure to grade the candidate's performance against the role.

Phase 4

Knowledge Depth

These questions really challenge a candidate's depth of knowledge on a particular skillset. For example, the questions might have candidates explain, in depth, how a specific technology works.

In fact, using the word "question" here is even a bit misleading. This should be a discussion that transcends more than just a list of questions that the candidate can get right or wrong. The hiring manager should know (with ease) whether this candidate has a deep enough skillset or not after this discussion.



Where To Use It

For roles that aren't entry level, it's best to explore the specific skillset that will help a candidate excel in a role that requires specific expertise. But, these questions should be used when you truly need a specialist. For example, you're building a new system and need someone who knows how to architect a scalable system or you're building a mail application and need someone who really know anti-spam algorithms.

Examples

These questions might sound like phase 1 and 2 questions, but the conversation surrounding them will be much deeper and open-ended.

- Explain how to use Map Reduce is, where you'd use it, and how it's implemented.
- Explain how garbage collection works in Python.

Where It Goes Wrong

This most often goes wrong when you didn't actually need a specialist and now you've unfortunately disqualified candidates that would have otherwise worked for the role. If you try to keep your performance expectation in other (often superfluous) areas too high, then you'll unnecessarily struggle with hiring because no one will be good enough. But, if you do need a specialist then you should never waver in these areas if a candidate's questions are inadequate. Elevating your expectations in superfluous areas or lowering your expectations in required areas can both be equally detrimental to your hiring. Thus, be very realistic and focused in this phase.

Phase 5 They're good, but are they great?

Ok, your candidates have met or exceeded your expectations so far and honestly they'd probably be a good hire. But, hiring a great engineer (versus a good engineer) can sometimes make a huge difference, even on a company's bottom line. If you hire an engineer that gives your website 20% more load efficiency then they just made you a little more money. But, hire an engineer that can re-architect your website to support 1,000% more traffic and they just made your company a lot more money!



Where To Use It

These questions should mix skill with creativity and be something no candidate could have easily prepared for nor simply regurgitate. What you're looking for here is creativity, outside the box thinking, and an ability of the candidate to examine a complex problem that they may not be familiar with, brainstorm something clever, and clearly articulate the solution.

Examples

- Our company is designing a lottery website. To simulate the lotto ball selection, design an algorithm to return a random node from a binary tree, such that all nodes are equally likely to be returned and also explain how you would safeguard this algorithm from outside tampering.

- We're launching a new social media engine. Design a system with TinyURL functionality that truncates a link to 12 characters (not including http://) but includes at least one dictionary word from the original URL that is shorter than six characters.
- A new initiative to reduce our cloud storage space has been green lit. Design a system that takes in files from millions of users which both minimizes the amount of storage used and maximizes file redundancy.

Where It Goes Wrong

Many companies attempt to do this but end up making one crucial error by judging the answer in terms of “right or wrong” only instead of evaluating the creativity of the answer as well. Maybe their thought process was brilliant but their exact solution (from a code perspective) would need adaptation. Don't be so stringent on the 'code' side that you disqualify a programmer who can think creatively and express his ideas articulately. This phase is definitely the hardest to get right but it's always worth its weight when done properly.

In Summary

Well, there you have it. With this five phase approach you can begin to master the hiring process for A+ programming talent. Remember to define what you want (and what you need) upfront and then follow this formulaic approach to end up with the strongest candidates to pick from at the end.



About Gayle Laakmann McDowell

Is the founder/CEO of CareerCup.com and the author of the Cracking the * interview series (Cracking the Coding Interview, Cracking the PM Interview, and Cracking the Tech Career). She now consults with companies on their tech hiring practices. She previously worked as a software engineer and hiring committee member at Google. She can be found online at: gayle.com | facebook.com/gayle | twitter.com/gayle



About HackerRank

HackerRank is a technical talent community for developers to hone their skills and for companies to hire best tech talent. With 1M+ developers, 35+ languages and 8+ programming domains, Hackerrank is giving companies recruiting tools such as codesprints and code challenges that make sourcing, screening and interviewing effective. 1000+ companies are revolutionizing tech recruiting with HackerRank.