

By Justin James

Programmers come with a wide range of skill sets, hail from many countries and cultures, and can have differing backgrounds and experiences. Nevertheless, certain qualities can mean the difference between a great programmer and someone who's not so great. Here are 10 things to look for when hiring a programmer.

1 Curiosity

Great programmers never accept things "as is"; they need to poke deep inside something, even when it appears to be working fine, to learn more. This is how many problems are solved before they *are* problems, and it's usually the quickest way to fix acute issues. A programmer without this mentality will usually end up lacking the knowledge underlying why they are doing what they are doing, which means they're working with blinders on. Unless candidates are very shy, their curiosity, if they have it, will show strongly during interviews.

2 Clear thinking skills

It may sound obvious, but programming is an exercise in logic. People who can add 2 and 2 to get 4 are common, but people who can take " $2 + x = 4$ " and figure out that "x" is equal to 2 are much less common. This is why I have always preferred programmers with strong math or science backgrounds. It makes them a bit better at programming, but more important, it generally indicates good logic skills. When I discuss the job, I sometimes leave blanks in what I'm saying to see if the candidate can fill them in. In addition, if your hiring process includes formal testing, that's a good time to test logic skills.

3 Top flight reading speed and comprehension

Another "duh" when it comes to programmer productivity is that most of their work is *not* the typing of the code. A significant portion of a programmer's day is spent reading, whether it be other people's code, Web sites with examples, documentation, or project specs. Programmers who read slowly, or worse, don't understand what they're reading, will be inefficient at best, and dangerous at worst.

You probably don't want someone on staff who misreads the spec and spends three weeks doing the wrong thing; that's just embarrassing when you need to explain the delay to the project sponsors. It's really hard to gauge reading skills during the hiring process unless you use a formal testing process.

4 Attention to detail

Attention to detail is a close cousin to curiosity. A programmer who pays attention to detail will be significantly more productive than one who doesn't, all else being equal. It is, unfortunately, extremely difficult to measure this quality during the hiring process. Still, sometimes things happen during the hiring process that show that a candidate has this trait. Maybe it's a casual remark or just a minor incident that occurs during the interview.

For example, I once had an interviewee casually compliment me on my shirt and mention that he was a fan of that designer; that spoke volumes about his attention to detail (as well as his fashion sense). And of course, a severe lack of attention to detail can sometimes be obvious too; the candidate who walks in with pants unbuttoned or toilet paper stuck to a shoe clearly is not paying attention to detail!

5 Quick learner outside of programming

Unless your company develops programming tools, like compilers and IDEs, your programmers are working with projects outside the realm of programming. Just as journalists need to understand a little bit about the subjects of their stories and good teachers need a working knowledge of the field they're teaching, good programmers are able to learn about the environment their software will work within. Of course, you don't need a CPA with a computer science degree to work on your accounting software, but a programmer who can't understand the basics of the math and business rules involved is going to be a liability.

I take candidates I'm seriously considering on a tour of the facility and provide a brief, simple, jargon-free overview of the company and how it works. Candidates who ask pointed questions that show they understand what I'm talking about, or who otherwise show comprehension, get extra credit in the overall hiring decision.

6 Self-learning skills

It's extremely rare for a programming shop to have the budget and time to provide training to its programmers. This is unfortunate, but it's a current business reality. The result is that most programmers self-teach their skill sets (ideally with a mentor handy) once their formal schooling is over. Programmers who are good at self-learning are going to be better at programming.

During the interview process, I like to ask questions such as, "How did you learn to do that?" when the candidate talks about something difficult, or, "How do you get new skills?" and "Do you read any programming-related books, magazines, Web sites, blogs, etc?". Candidates who aren't just capable but who are eager to teach themselves new programming skills are much better to have on staff than those who don't like to learn outside a formal training program.

7 Passion

Some programmers are "daycoders"—people who write code 9 to 5, Monday through Friday, and do not think about it in the slightest outside of those times. That's perfectly fine; not everyone can be a super geek who lives and breathes code. I have hired people like this in the past to fill a gap or to work on the sections of a project that are routine. But when I need to hire a *top* programming candidate (regardless of skill or experience level), I need to be hiring someone with a passion for the work.

Passion is a "make or break" during crunch time or on a project that requires tricky techniques, rare skills, and so on. After all, daycoders won't be motivated to learn the best way of doing things and will instead just do what they've always done, which may not be the best way of doing things. Daycoders are also difficult to retain without a steady stream of raises and a high level of perks, since they are there for the money, not for the work. Passion will be fairly obvious during the interview. Candidates who get excited when you talk about your project or who are talking about their past projects have it for sure.

8 Adaptability

Have you ever worked on a programming project that ended with the same specs it started with? Neither have I, and I am including short projects that lasted less than a day! Programmers who do not handle change well will probably not be very successful, except on long-term, Waterfall-type projects that last years, usually under government contract. That is not to denigrate those kinds of projects or programmers, of course. But most projects are simply incompatible with a lack of adaptability.

It's pretty obvious during interviews when candidates are not adaptable or handle change poorly, particularly if you ask questions like, "Did the requirements change often?" Candidates who say something like, "Sure, but that happens on all projects and it's a fact of life" are winners. Those who roll their eyes and respond with, "Yeah, that's why I could never get anything done!" probably will not be a good fit for most environments.

9 Good communication skills

"Communication skills" doesn't mean the same thing as "Speaks perfect English." It means, "able to convey an idea accurately and effectively." Pictures, sounds, and hand motions are all part of communication skills. Programmers who have a hard time getting their point across or understanding what others are trying to tell them will not be effective in the long run. This is a difficult ability to properly measure in a phone interview, but when candidates have difficulty communicating even in a face-to-face interview, you can be sure that they'll have a hard time on the job as well.

10 Who's the boss?

Programmers are a notoriously independent group of people. Indeed, I believe that's one of their strengths, and it's great not having to micromanage people working on technical projects. However, a good portion of programmers struggle with the idea of "I am the boss and you are not." I know, it sounds tyrannical. In a way, it is. Managers often need to make decisions for nontechnical reasons, and they may not be able to explain those reasons to their team (secrecy, politics, not enough time, etc.).

A little bit of pushback, particular on bad decisions, is something I encourage and fully support, especially if the boss doesn't realize that it is a bad decision and if the feedback is delivered correctly. But when the boss says, "I know from the technical perspective this is a bad idea, but this is how we need to do it," it's final. All too often, certain "rogue coders" will ignore their marching orders and go do their own thing. Even worse, they have a tendency to run their mouth to anyone and everyone about how stupid the boss is, and how he or she obviously does not understand programming—which may or may not be true. This sinks projects and does nothing but cause animosity and hurt team morale.

This mentality can often be seen during the interview process, especially when you're asking about past work experiences. Rogue coders love to talk about their "evil, idiot, pointy-haired slave driver" former managers, even when it is wholly inappropriate to do so, like in a job interview. Well-adjusted programmers will say things such as, "I disagreed with some of my manager's decisions at a technical level, but I know that those decisions had to have nontechnical issues factored into them."

Additional resources

- TechRepublic's [Downloads RSS Feed](#) **XML**
- Sign up for the [Downloads at TechRepublic](#) newsletter
- Sign up for our [Software Engineer newsletter](#)
- Check out all of TechRepublic's [free newsletters](#)
- [10 signs that you aren't cut out to be a developer](#)
- [10 tips for writing a job-winning developer resume](#)
- [Steer clear of these 10 illegal job interview questions](#)

Version history

Version: 1.0

Published: March 5, 2008

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to [drop us a line](#) and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Content Team