* Why did you leave your last job?
* What are your career goals?
* Why did you want to be a software engineer?
* What are some key projects you worked on?
* Are you a team player?
* Why should we hire you over the other job applicants?
* How well do you deal with stress?

**1: Tell us about your current position**

Employers want to know about what you are currently doing a lot more than they want to know about prior positions. The reason for this is simple: The world of software development moves so fast that what you did two or more years ago is interesting for background but probably has little bearing on their current work. When asking this question, the interviewer is trying to relate what you currently do to the position the company is offering, and you will want to answer with that in mind. For example, if the position you are applying for involves a lot of database programming, emphasize where in your current job you have worked with databases.

**2: Programming challenges**

Many employers will present you with some sort of programming challenge. These range from being asked to sketch out a piece of pseudo code that implements some business logic or being handed a piece of code and told to find the bugs to being put down in front of a computer and asked to code away. What they are usually looking for is not just a certain level of competency -- they also want to see how you go about solving the problem. You can offer to narrate your thought process as you solve the problem. If they take you up on it, that will help them to learn what they are looking for. Or perhaps when you are done, you could walk the interviewer through how you solved it.

**3: Do you have any examples of your work?**

Employers love to be able to look at real-world examples of your work. Unfortunately, this is rarely possible. The truth is, in most circumstances, your work is the property of your employer and you can't be taking it outside of the building without permission. And it would be unusual to have a boss say, "Sure, go grab a couple of your best apps from source control to take on the job interview!" Instead of being unable to provide any samples, contribute to an open source project or work on an application at home that is sophisticated enough to let your skills shine. Then you will have something that you can show the interviewer and also be able to demonstrate an ability to work on your own and manage your own time, too. These side projects can often serve as a great talking point in the interview.

**4: Brainteasers**

Apart from asking you to demonstrate some programming abilities in the interview, some employers may give you a variety of brainteasers. Some people are really good programmers and stink at these, but the idea is to test your overall problem-solving skills. Luckily, you can prepare for these a little bit by picking up a few brainteaser books (usually only a dollar or two) at a book store or supermarket and doing a few every day. Most of these brainteasers follow a similar format, so by practicing, you will understand how to approach the most common types. There are also a few standard ones that come up on a regular basis, such as the one where [you need to get a group of people across a river](http://brainden.com/crossing-river.htm) with a boat of limited capacity.

**5: Do you have a security clearance?**

Depending upon the job, a security clearance may be required. Employers prefer hiring people with one already because it simplifies things. It would be a big hassle to hire someone and then discover that they can't get the needed clearance to do the job. If you have a clearance, make sure that it is up to date. It's also a good item to list on a resume.

If you do not have a security clearance, ask before you come in for the interview about any security requirements for the job and research whether you are eligible for any security clearances needed. This way, when asked, you can answer with something like, "No, I do not have that clearance, but I have looked into it and I can obtain one if needed."

**6: Background check and criminal history information**

Information about criminal history and other background check items typically will not come up in an interview with a hiring manager, but they will often come up in an interview with HR or a recruiter (especially the recruiters). They do not want details, for the most part, but they want to know whether it will be a waste of time interviewing you. Obviously, it is great to have a squeaky clean record, but there are plenty of good job candidates who don't. You will need to be honest here, because things show up on the background check anyway. If what you say does not match the check, they will feel that you lied to them. At the same time, limit your sharing to the minimum. You can start with something like, "I have a misdemeanor conviction from three years ago" and take it from there.

**7: What is your experience level with XYZ?**

When interviewers ask about your experience level with a technology, they really want to get a feel for what you have been doing with it, not how long you have been doing it. For example, if they are asking about SQL, is it important to them that you have been writing statements no more complex than, "SELECT id, name, city FROM people WHERE state = 'NY'" for 10 years? Not really. Performing complex data transformations, correlated subqueries, etc., for six months will be much more impressive. When talking about your experience level, emphasize the kinds of challenges you solved with those technologies and the unique aspects of the technologies you used to solve the problems.

**8: What's the hardest challenge you have had to overcome -- and how did you approach it?**

This is a stock interview question, but it has some special pitfalls for the programmer. One of the failures I've seen in interviews is that candidates do not properly set the context of their answer. I have faced some challenges that at *that point* in my career were difficult but that later on were trivial. If I brought them up in an interview without explaining my experience level when they arose, it would put me in a bad light. The interviewer would be thinking, "Why would someone with his experience struggle with this?" So when you answer, give a short (one sentence) scene setup. Also, put your focus on the problem-solving steps you took, not the technical details. No one really cares if the problem turned out to be that the variable was one character shorter than the data going into it; they want to know how you did the research to discover it.

Talk about my SQL crash back in 09ish?

**9: Describe your programming habits**

There are a number of variations on this question, some of which just ask about things such as:

* Source control
* Testing
* Variable/file/class/whatever naming
* Application architecture decisions

Some things we do by habit are not flattering when we answer these questions, but it is because of circumstances outside of your control. For example, if your current employer does not have a source control system, do not say, "I do not use source control" because it makes you look awful! Instead, an answer such as, "My current employer does not have a source control system, but I have used TFS at a previous employer, and I use Mercurial at home for personal projects" will be much better.

Other times, we simply have bad habits; in those cases, it is better to recognize them and show that you are trying to change them. You could say something like, "I tend to not write as many unit tests as I should, but I have been working hard to ensure greater code coverage." Of course, don't lie about this. But employers like to find people with enough self-awareness to see and correct their flaws, and the honesty to be able to discuss them.

**10: Tell us a little bit about yourself**

Often, job candidates go way off the deep end on this question, talking about things they do not need to be discussing in a job interview, personal stories and relationships, and so on. Or worse, they bring up things that present them in an unflattering light. What the interviewer is really looking to learn is how your personality relates to the job of *software development*. For example, if you enjoy restoring antique furniture, you could point out that it requires a lot of patience, eye for detail, research, and so on. Of course, you will want to talk about your personality traits as well. Unusual experiences or education can be brought up here, too. What you definitely do not want to do is talk too long. Try to make it a back-and-forth conversation, but if it isn't, limit your time to a few minutes and don't trip all over yourself trying to cram in every last detail.