# **The Complete Software Developer’s Career Guide**

**JOHN SONMEZ**

**Chapter 11**

**Getting a Job Without Experience**

When you don’t have any experience, it can be extremely difficult to get a job.

In the last chapter, we talked about one way to get around this through the internship process, but that isn’t always an option.

No need to lose hope, though. If you can’t get an internship and you don’t have experience, you can still get a job—and I’m about to show you how.

This chapter isn’t going to be about getting the job so much as preparing yourself to be able to get a job as an aspiring software developer without any, or with very little, experience.

In the next chapter, we’ll go into the details of how to best apply for and find jobs—whether you are experienced or not.

Before you start applying, you need to make yourself as marketable as possible and that is what this chapter is all about.

**The Biggest Risk for Companies Hiring a Developer**

Let’s start off by talking about the biggest risk factor for most companies when it comes to hiring a software developer.

Do you know what it is?

Well, having been on both sides of the table multiple times and being a business owner myself, I can say that in most cases it’s hiring someone who doesn’t actually know how to code.

A software developer who doesn’t know how to code—or is really bad at it—actually adds negative value and can cost the company even more money than merely paying their salary would entail.

In such a case, it’s better to never hire this person at all.

When I worked for Hewlett Packard, part of my job was to interview “top-notch” C++ programmers to add to our team. These programmers would join my team of proven top-notch C++ developers whose job it was to debug the most complex problems and triage them to be fixed by the product development team.

Oftentimes, I’d sit at the interview table and look at a resume of a supposedly “expert C++ programmer” who had over 15 years of experience, and I’d ask them one simple question about C++, but they couldn’t answer it.

One simple question.

I’d ask them to write some code, and [they’d have 100 excuses](https://simpleprogrammer.com/cg11-excuses) for why they couldn’t write code right now or why their code to a simple problem didn’t work.

These were supposed to be expert developers.

Remember, these people had excellent resumes, made it past HR, made it past a technical phone-screening interview, and were now sitting in front of me, desperately trying to bull shit me into believing they actually knew how to code.

Why do I bring this up?

Because I want to emphasize that any good technical interviewer has seen it all before and has hired enough bad programmers to know that their primary job is to prove that you don’t actually know how to code.

Hey John

Are interviewers really out to get you? Do they really want to see you fail?

So… I do a bit of dating coaching for guys mostly.

One of the things I coach guys on is how to approach a woman that he might be interested in and how to start a conversation.

I always tell these guys that when you approach a woman, say at a bar, most of the time it seems like she doesn’t want to have anything to do with you and finds you annoying. She gives the impression that you need to get lost.

But, secretly, most single women are really hoping you’ll be “that guy.”

That awesome guy who is the man of her dreams—confident and self-assured and does what he thinks is right, disregarding the disapproval of her or anyone else—a manly man, to be politically incorrect.

I tell these guys that the reason a woman may act as if you are annoying or that she wants nothing to do with you is because she’s testing you.

She wants to see if you really are the confident man that approached her or if you are just pretending to be that guy and somehow mustered up the courage to say “hi,” while at the first sign of disapproval you’ll crawl back into your hole and cry yourself to sleep.

It’s the same with an interview.

A good interviewer really, really wants you to succeed.

They want you to be “that guy” or “that girl.”

Believe me, they don’t want to keep having to interview and turn people down, but just like that pretty girl at the bar, they have to test you—make sure you aren’t full of shit.

So, if they are smart, they are actually going to try to prove that you don’t know how to code.

It’s going to seem like they want you to fail—even though they secretly want you to succeed.

Be prepared for that, don’t let them rattle your cage.

Keep your cool, be cool… see if you can get her to buy you a drink.

Any company that you would want to work for is going to guard as much as possible against hiring developers who don’t know how to write code.

This is one of the reasons why—even though so many software developers complain about it—so many companies do [whiteboard coding interviews](https://simpleprogrammer.com/cg11-whiteboard). (It’s really difficult to bullshit one of those.)

**Overcoming That Risk**

Where am I going with all this?

Simple. If you have no experience, your primary strategy to getting a job as a software developer is to prove that you can code and that it’s not BS.

Everything else I’m talking about in this chapter is going to be based on the idea that you want to alleviate that risk as much as possible.

When it comes to interviews, [not everyone is as hard on potential developers as I am](https://simpleprogrammer.com/cg11-hard), but I can tell you that even the dumbest interviewer will be at least somewhat suspicious of the abilities of someone who doesn’t have any programming experience at all on their resume.

In fact, you probably won’t get an interview at all unless you can figure out how to instill enough confidence in your coding ability or competence to be able to overcome that severe handicap in the first place.

So, what we really want to do here is make it pretty clear to any company considering you for a position that, even though you don’t have direct work experience, you are an experienced programmer, and you have some external proof that you know how to code.

That is what is going to help you get your foot in the door.

**Have an Online Presence**

First of all, you should have an online presence of some sort.

One of the first things a potential interviewer is going to do when you are being considered for a job is to Google your name.

In fact, the HR person who got your resume is probably going to do this even before you get close to an interview.

It’s going to be really good if what comes up is your professional blog where you have a bunch of articles about the specific software development technologies you are specializing in, and it looks like you keep it up-to-date.

Every software developer should have a blog.

It doesn’t matter if you are an experienced developer or if you are just starting out. You should share what you are learning and write about your chosen specialty in the software development field.

I’m not going to go into the details of creating a blog and specializing here, but I’ve heard every excuse you can possibly think of as to why people don’t have a blog. So let me say, regardless of experience, regardless of whether you think you have nothing worthwhile to say, just do it.

At the very least, a blog will show that you have some passion and dedication to your interest in software development and that you are the kind of person who likes to learn and help others.

Aside from a blog, it also helps to have other good things come up when your name is Googled.

A Twitter account, perhaps a Facebook page, or something else that shows you are active and involved in the software development community is going to go a long way in helping you establish credibility, despite lacking experience.

I’d highly encourage you to Google your own name and see what comes up.

In today’s world, this is your real resume, regardless of what kind of resume you create.

**Have a Portfolio**

Having an online presence is great, but without experience, you are probably going to want to have a bit more substance as well.

It’s a really good idea to [have a portfolio of some of the work](https://simpleprogrammer.com/cg11-portfolio)you’ve done that shows you know how to code and provides an example of your code.

This is going to go a long way to alleviate that risk a potential employer would have in hiring someone who lacks experience.

If a potential employer can see code that you’ve written and projects you’ve created, they can be more sure that you at least have some idea of what you are doing and can probably write code for them.

Now, there isn’t any way to prove that you wrote the code yourself or that it is even your code, but if you have a decent portfolio of projects you created, it’s more and more believable.

I’d highly recommend creating some small projects while you are learning to code, or after you’ve learned to code, and demonstrate your ability by writing an entire application from start to finish.

You can put these projects online using a service like [GitHub](https://simpleprogrammer.com/cg11-github).

In fact, it’s worth taking a moment to talk about GitHub, which can act as an online portfolio and is already used to judge, to some degree, the ability of even experienced programmers.

GitHub is an online and open repository for many code projects, especially open source ones, based on the source control system, Git.

But today, it’s much more than that.

Many software developers are judged by their GitHub profile, which shows [what projects they committed code to](https://simpleprogrammer.com/cg11-open), what projects they’ve created, how often they commit code, and how popular their code is.

Using GitHub is one of those ways to prove you have ability when you lack actual employment experience.

If I were starting out without experience, I would make my GitHub profile as impressive as possible.

Your GitHub profile can essentially act as an online portfolio for your work.

Even if you don’t use GitHub, though, you should still have a portfolio of your work.

Another great way to create a portfolio is to create actual mobile applications that you can deploy into one of the mobile app stores.

It’s really easy for anyone to do this today, and there are multiple benefits beyond just creating a portfolio for yourself.

You could also make some money, and you might even be able to build your own side business—more on that below.

I’d aim for having a portfolio of at least 3-4 applications or projects you’ve built.

They don’t have to be big, but they shouldn’t be completely trivial either.

Try to demonstrate your skill in the technology you are striving to get a job in. It also helps to show some coding ability to do something like call a web service or utilize a database.

You might also want to include unit tests or automated tests of some sort to show that you have the ability to write test code.

You will want to show off the best of your abilities with these sample projects in your portfolio.

It can also be a good idea to bring your laptop into an interview with one of these projects already loaded up so that you can show the interviewer the code you wrote and go through why you designed things the way you did and how you created one of the applications in your portfolio.

Honestly, I would not try to be a software developer today without some kind of portfolio. I think it is the single best way to demonstrate your proficiency because, for the most part, it’s incontrovertible proof that you know what you are doing.

**Create Your Own Company**

Many people laugh when I tell them this idea of gaining experience when you don’t have any, but it’s perfectly legitimate.

Way more companies than you probably realize are actually run by a single person or a skeleton staff of part-time workers or contractors.

There is absolutely no reason why you cannot create your own software development company, develop an application, sell or distribute that app, and call yourself a software developer working for that company.

You can do this at the same time you are building your portfolio and learning to code.

If I were starting out today, I’d form a small company by filing for an LLC, or even just a DBA (Doing Business As) form (you don’t even need a legal entity), and I’d build an app or two that would be part of my portfolio. Then, I’d publish that app or apps in an app store or sell it online in some way.

I’d set up a small website for my software development company to make it look even more legit.

Then, on my resume, I’d list the company and I’d put my role as software developer.

I want to stress to you that this is in no way lying and it is perfectly legitimate. Too many people think too narrowly and don’t realize how viable and perfectly reasonable of an option this is.

I would not advocate lying in any way.

If you build an application and create your own software development company, there is no reason why you can’t call yourself a software developer for that company and put that experience on your resume—I don’t care what anyone says.

Now, if you are asked about the company in an interview, you do need to be honest and say it is your own company and that you formed it yourself.

However, you do not need to volunteer this information.

I don’t think being the sole developer of your own software company is a detriment either.

I’d much rather hire a self-starter who formed their own software company, built an app, and put it up for sale than someone who just worked for someone else in their career.

I realize not all employers will think this way, but many will. You’d probably be surprised how many.

In fact, Simple Programmer literally started this way.

**Prepare for Interviews**

Anyone looking for a job should prepare for interviews, but someone without experience should especially do so because you are going to be judged much more harshly, and you are going to have to answer many more tricky questions.

So, do your homework. Spend plenty of time preparing for interviews by both studying questions you are likely to be asked and doing mock interviews with friends, relatives, or whomever is willing to help you.

Get a camera and record yourself doing an interview.

Play it back and watch to see what you look and sound like.

Go get a book like [Cracking the Coding Interview](https://simpleprogrammer.com/cg11-cracking) and make sure you can pass any kind of coding interview you get.

You are going to have to really prove yourself, so you need to be extra prepared.

When you have no experience, you are starting off with a distinct disadvantage, so you will have to work extra hard to overcome it in an interview.

**Build Your Network**

This is another one of those things that anyone looking for a job should be doing, but you especially need to do this if you don’t have experience.

The best way you are going to get an opportunity without experience is if someone is willing to take a shot on you and give you a chance because either they know and like you or they know someone who will vouch for you.

[Make sure that you are attending community events](https://simpleprogrammer.com/cg11-community) like meetups or developer organizations.

Work hard to build up a network with people who work for various companies you might want to get a job at.

Again, this is just another way to overcome the disadvantage of not having experience.

If I were starting out without experience, I’d be spending extra effort building my networks.

**Offer to Work for Free**

I’m slightly getting into tactics here, since this will apply mostly for someone without experience and I’m not going to cover this in the next chapter on finding a job, but I don’t want to leave this one out.

One really good way to eliminate risk for a company that is considering hiring you when you don’t have any experience is to basically offer to work for free or—even better—offer a money-back guarantee on your work.

I know this sounds crazy—and I admit, it’s a bit difficult to pull off.

You are probably not going to take this approach when you go through the normal resume / interview process, but if you are working through your network or reaching out directly through an informal interview process, you might really want to consider this tactic.

You do have to have [a lot of confidence](https://simpleprogrammer.com/cg11-confidence) to pull this off, though.

You have to project the idea that you are so sure that you can succeed that it is well worth taking a chance on you and that you are even willing to work for free or with a money-back guarantee to prove it.

You’ll have to be fairly persistent and you’ll have to really think outside of the box—like I said, this isn’t going to likely work in the typical resume / interview process. However, if you are an out-of-the-box thinker and you have an affinity for salesmanship or [very high charisma](https://simpleprogrammer.com/cg11-charisma), you can make this work.

In fact, if you pull this off confidently enough, you may even get hired without having to work for free or offer a guarantee. Just making this kind of offer can give a prospective employer enough confidence in your abilities that they are willing to take a chance on you and hire you straight up.

Like I said, this is a long shot, but I have heard multiple stories of software developers successfully pulling off this tactic.

Besides, if nothing else is working and you are desperate, what have you got to lose?

**Offer to Work on a Small Project**

If offering to work for free or with a money-back guarantee seems too bold, cocky, or risky—or perhaps all three—there is a smaller version of the same idea which can help reduce risk for an employer by giving you a chance to prove yourself before being hired.

You can always offer to work as a contractor or consultant on a very small project first in order to demonstrate your abilities.

Again, this is going to require at least some out-of-the-box thinking, but if you pitch this right—especially to a company that “is not hiring right now”—you might be able to pull it off.

In fact, many companies hire employees as temporary contractors first to “test drive” them.

Another variation of this same tactic is to gain some experience by doing some really cheap freelance work.

You can sign up on a site like [Upwork.com](https://simpleprogrammer.com/cg11-upwork) to work as a freelancer and bid on jobs.

If you are willing to work for a really low rate, you might be able to get a job even without experience.

Perhaps you would expect to make $25 an hour doing a programming job. You could always offer to do work at $5 an hour just so you could gain some experience.

Project some confidence. Say that you would normally charge $25 an hour and that you can do as good of a job as someone who charges that rate or higher, but you are trying to gain some experience, so you are willing to work at a much more discounted rate.

I’ve even hired freelancers on Upwork who have clearly lacked the experience, but their rate was so low and they were so eager to get experience that I decided to give them a shot.

At a very low rate, I figured I didn’t have much to lose and if they turned out to be good workers, I’d be getting a hell of a deal.

It doesn’t have to be Upwork, either. I did an [interview with Marcus Blankenship](https://simpleprogrammer.com/cg11-marcus) and he talked about how, when he first started out doing freelance web development work, he worked for a ridiculously low rate just to gain experience.

As long as you present your offer effectively and you are willing to take a big pay cut, you can trade a lower paycheck for experience—which will be much more valuable in the long run.

**Work in the Mailroom**

I don’t always advise the mailroom tactic because it can get you stuck in a role that you don’t want to be in, and it can be difficult to transition into a software developer position in certain organizations. Sometimes, though, this is the best choice when you lack experience and you need to get your foot in the door.

Earlier in my career, I had very little experience and the hiring craze of the dot-com boom had just ended.

It was sort of difficult to get a software development job.

I tried applying to many different jobs and I even had several interviews, but after three months, I didn’t have any real prospects.

Finally, I decided to call a buddy of mine who worked at HP and see if there was any way he could get me back in there.

He said that there were no software development jobs he could help me with but they needed a QA person, and he was pretty sure he could get me the job if I wanted it.

I knew that I didn’t want to be in QA, but I felt like once I was working at HP, I’d be more likely to get a software development job there eventually, so I took the job.

It didn’t take me very long before I started helping out one of the programmers that was sitting near me, and after some time I was moved into a software development role and eventually promoted.

Even though it might not be ideal, you could start in another position in a company and work your way up to a software development role.

There are, of course, challenges in doing this, and it is difficult to get someone to change their perspective of you if you come in as a certain role and now want to transition to a software developer. Nonetheless, if you are lacking experience, this technique can work.

It will at least help you get your foot in the door.

**Get Certifications**

[I’m not too big on certifications](https://simpleprogrammer.com/cg11-certification), but I do think they become much more valuable when you lack real job experience.

Getting certifications won’t prove you know anything and it won’t guarantee you a job, but it is another way to alleviate some of the uncertainty a prospective employer may have with hiring someone who lacks experience.

At one point in my career, I lacked .NET experience, but I really wanted to get a role doing .NET development because I liked C# and thought it was the future.

The problem was, even though I had some C++ experience, no one wanted to hire me for a .NET job—especially since I didn’t have a college degree at the time, either.

So, what did I do?

I got every single .NET certification I could.

I got my MSCD, then my MCAD, and even my MCDBA for good measure.

I had just about every developer certification you could get from Microsoft.

Within a year, I was moved into a .NET position, despite any real-world experience using the technology.

Now, like I said, this might not work for you, but I don’t think it can hurt—especially when you lack experience.

**Be Persistent**

Finally, remember that the squeaky wheel always gets the grease.

I used to put at the bottom of my email signature, “I am the squeaky wheel,” to remind everyone that I’m going to keep coming at you until you give me what I want.

[I owe my success in life more to being persistent than anything else](https://simpleprogrammer.com/cg11-persistence).

When you don’t have experience, you should make up for it with gumption.

Be the kind of go-getter who doesn’t take “no” for an answer and people will be likely to eventually give you a shot—experience or not.

Many people are afraid of following up for fear that they’ll blow their chances or be annoying.

Be annoying. It’s better than being forgotten.

Besides, if you are persistent enough and you do it in a way that is as least annoying as possible, you’ll come across as someone who is eager and hungry.

I’ve been on both ends of this.

I’ve followed up with 10 emails before getting a favorable response from someone I was trying to get in contact with.

I’ve also received emails asking for my time which I’ve blatantly ignored until the sixth or seventh one made me take notice and realize that someone this persistent might be someone worth spending some time with.

If you don’t have experience, you are going to have to get it somehow.

Don’t give up. Keep following up. Keep doing what you are doing, and eventually you’ll find your break.

## Chapter 20

### How the Recruiting Industry Works

I can remember the first time I started working with a recruiter.

I had no idea how the recruiting industry worked.

I took my freshly created resume, sent it to a recruiter, and said, “Find me a job.”

It didn’t go so well.

I kept calling day after day, asking the recruiter if they had made any progress on finding me a job.

I was under the false impression that “my recruiter” worked for me.

It was only after some time and quite a few negotiations and placements that I realized recruiters—at least for the most part—don’t work for you. They work for the company who has the job opening and has hired the recruiter to fill it.

(There is one exception, which I’ll talk about a little later on.) I know this may seem obvious to you, but I was oblivious to it. I just wanted a job.

In the years I’ve worked as a software developer and coached many other software developers to find and get the best jobs, I’ve learned quite a bit about how the recruiting industry works.

It’s actually pretty crazy.

But while it’s crazy, it’s also important that you know how the industry works so you can better navigate the choppy waters of getting a software development job.

It’s very easy to be taken advantage of, led on, and manipulated if you don’t understand this complex industry.

In this chapter, I’m going to share with you everything I know about the recruiting industry and give you some advice on how you can effectively apply this knowledge to get a good job and not end up with the short end of the stick.

Let me mention a couple of caveats before we get into it:

First, I’m not a professional recruiter, so most of my information is from what I have learned and observed.

Second, the recruiting industry is complex. I’m sure more than an entire book could be written on the topic. Here I’m simplifying, generalizing, and giving you a summary.

This is in no way meant to be a comprehensive description of the entire recruiting industry.

Still, I think you will find the information in this chapter useful and surprising.

### Types of Recruiters and Agencies (and How They Get Paid)

Recruiters go by many different names: headhunters, employment agencies, staff augmentation, those annoying guys who send you unsolicited LinkedIn requests for jobs you aren’t even remotely qualified for.

However, not all recruiters and recruiting agencies are the same—and not all of them get paid the same way, either.

Understanding each type of recruiter and how they get paid can help you to understand how to work with them—and even which ones to avoid.

Let’s break down each of the major types of recruiters, how they make their money generally, and what this means to you.

### Independent and Small Recruiters

These are typically the most annoying recruiters.

Some of these recruiters work on their own. Some of them work on some kind of heavily weighted commission structure and are actually contracted out en masse by larger firms with a sort of scattershot approach.

Since these recruiters or small agencies don’t have a large presence and reputation, they have to aggressively solicit potential candidates to try and fill a position.

Why are they so aggressive, you may ask?

Ah, it’s all in how they are paid.

You see, most recruiters get paid a commission based on a percentage of a candidate’s salary.

Before I tell you how much, go ahead and take a guess to see if you can figure it out—the answer may surprise you.

Did you guess?

In general, recruiters are paid between 20 to 35 percent of the annual salary of a candidate they place.

As you can imagine, in the software development industry—especially with recruiting for top jobs—this can be a large amount of money.

Is it all beginning to make sense now?

Can you see why recruiters would be so aggressive and fill up your LinkedIn inbox with messages?

Independent recruiters are often the most aggressive because they don’t have to share their commission with an agency they work for, or they get a much larger chunk of it.

Let’s use some actual numbers, so you can understand how powerful of a motive it is for a recruiter to place someone.

Suppose a recruiter is hiring for a Senior Software Engineer position and the salary is about $100,000 a year.

An independent recruiter is likely to get a 25 percent commission on placing someone in that job.

So $100,000 \* 25% = $25,000.

That is a pretty large score.

Now, like I said, recruiters usually aren’t completely independent, so they won’t get the full $25,000, but some are and do get that full amount. The ones who work for smaller agencies, where they are more like contractors themselves, will take a large chunk of that commission.

However, that is not the only way they operate.

Recruiters can also recruit for contract positions. (See the previous chapter on contracting versus salary for more information on contracting.) For these positions, recruiters place a candidate and then get a markup on what that candidate is being paid.

Let’s suppose a company is looking for a contractor and is willing to pay $75 an hour.

A recruiter might offer that job at $50 an hour, even though the company is paying $75 for every hour that contractor works.

In this case, their markup or margin would be $25 an hour.

I’m sure you can see how this would be quite lucrative as well.

### Large Agencies

Large agencies are going to operate in a structure similar to small agencies and independent recruiters with only a few differences.

The biggest difference is going to be reach and how commissions are handled.

Large recruiting agencies will likely have good relationships established with major employers, so they will be much more effective in getting someone placed than a smaller recruiting agency with fewer connections—at least from the perspective of the job searcher.

Often, large agencies can have enough of a relationship with a large employer that they almost take over the hiring process from the employer, and if they deem you a good fit for a job, you are very close to getting the job already. (This mainly applies for contract positions.) In those cases, the only way to get in might be through that agency.

With a larger agency, as might be expected, the recruiter takes less of the commission.

It’s likely the actual recruiter may get 50 to 60 percent of the recruiting free with the rest going to the agency.

There may also be a split with an account manager who handles the big corporate accounts for the recruiting agency.