

---

# Getting a Gig: A Guide

## Introduction

Hey friends! This is a guide for getting a gig in college as a tech major. I graduated May 2014 with 10 job offers after having 5 internships throughout my four years of college. You can do that, too. Because you're great.

## Contents

- Introduction (you read that already)
- Your Resume
- Your Cover Letter
- Your Attitude
- Your Skills
  - Building Them
  - Selling Them
- Your Search
  - Events
  - Networking
  - Referrals
  - Cold Calling
  - Fellowships
- Conclusion

## Your Resume

Your resume is your personal summary sheet. Your resume is the thing that gets your foot in the door. So, there's a few things you should do (and not do) to make it as awesome as you are.

### **Make your name BIG.**

Your name has to stand out from everything else, because you want it to be remembered. Making it the biggest thing on the page is the easiest thing you can do to make that possible. I've seen soooo

---

many resumes where the name is at the top, but it's just bolded and centered and they expect that to be enough. It's not.

### **Remove the objective.**

Nobody looks at the objective. Nobody. Unless it's unusual. I personally spoke to a bunch of recruiters from various companies and they all said that they never look at them if it's already relevant to the position you're applying for. If you're saying something like, "I want a part-time job so that I can do operatic bird training for the rest of a given week," yeah, that's worth putting an objective in, I guess. But if not, use that space to talk about projects you've done, activities you've done, etc.

### **Keep it to a single page.**

Recruiters are looking for a short summary of you. They're reading several resumes a day, and if they see something longer than they typically read, they could pass over yours for something more concise! If you'd like to say more, put a link to a personal website or portfolio for someone to find it. A resume is a summary, not a tome.

### **Remove irrelevant information.**

I know that lifeguarding in high school was a good gig that helped you gain people skills and attention to detail. But you're in tech. That doesn't matter as much to tech companies. Sorry, buddy. I still think you're great with people. When you're a first semester freshman, it's okay to have old high school stuff on there, just because it's less likely that you have other things to put on your resume. But as soon as you have a college GPA to work with, a club or two, and some volunteer experiences to replace that, do it.

### **Don't make it a scavenger hunt.**

When an application reviewer (engineer, recruiter, or otherwise) is looking over your resume, don't make it difficult for them to understand who you are and what you know.

For example, if you have online profiles like GitHub, LinkedIn, Twitter, or even your own personal website, put it on your resume. Don't make them have to search long and hard for you online if they want to know more!

If you decide to put relevant coursework on your resume, **please**, don't just put course numbers. Nobody knows what that means. And nobody is going to go to your university's website to find out exactly what CS229 is. Put down the course titles instead!

---

And finally, put down your graduation date. So many students I've seen don't put it on there because they are hiding the fact that they're a freshman, or they're "technically a junior if you count the credits." That doesn't matter. Trust me. Just put down your graduation date so that the company knows how much experience you have, and how soon they can potentially pull you in for full-time.

### **Include only certain personal information.**

Companies aren't allowed to ask about your religion, marital status, or race/ethnicity, so you shouldn't include that.

In terms of contact information: you don't need your mailing address. That is a thing of the past. Just like my youth. Tech companies email, and **maybe** call. That's all you need! Some great things that you might also want to put on there are your personal website (if you have one, which you should), your GitHub profile (if you have one, which you should), and your LinkedIn (if you have one, which you should).

Though there's some who might disagree, you should **include your GPA**. If your major GPA is significantly higher than your overall GPA, you might want to consider having both. When you have graduated and have a full-time job, you won't need it as much. But at this point in your beautiful life, keep it in there. Some companies care about GPA more than others, so you might as well play it safe and have it there.

## **Your Cover Letter**

Your cover letter is your written sales pitch. You've got a resume that summarizes everything. Now you have to write out a more complete, professional description of you and what you can offer a given company. Here's a sample cover letter to get you started:

Dear \_\_\_\_\_,

I hope your day is going well! My name is \_\_\_\_\_, and I'm a \_\_\_\_\_ at \_\_\_\_\_. I am very interested in working for \_\_\_\_\_ next \_\_\_\_\_. Your commitment to \_\_\_\_\_ and \_\_\_\_\_ that I saw on the website inspired me! The products you build and the values you stand for make \_\_\_\_\_ seem like an ideal workplace for me. A little about me, I [insert relevant work experience, extracurriculars, and projects here]. I think these experiences would make me a great candidate for you.

Please let me know if there's anything else you need from me. I look forward to hearing from you! I can be reached at \_\_\_\_\_ and \_\_\_\_\_

---

Best regards, \_\_\_\_\_

Now, remember, this is just a sample. You can write a cover letter in any format you'd like. But, you should be sure to include the following:

- **Who** - Who you are. Easy enough.
- **Where** - Where you're coming from.
- **Why** - Why you're interested in this company, and show that you researched them.
- **What** - What you can bring to the table.
- **When** - When you're available to start, and when they can contact you.
- **How** - How they can reach you.

## Your Attitude

When you're internship/job hunting, it's very easy to feel down if you don't hear back from companies, an interview goes poorly, or you start comparing yourself to people. It's a tough field we're going into.

So. I won't spend much time on this. But I want to emphasize something: **You're brilliant.** You're in this field for a reason. When your grades aren't awesome or someone gets something that you wanted, don't feel down on yourself. So many opportunities will come your way if you just keep working hard and refining your skills.

Mark Twain once said, "Comparison is the death of joy." When you start to compare your skills to others, it's hard to not feel as good about your own, or to get a little too competitive about your work. Work hard and don't let others get you down. It's remarkable how significantly that can improve both your work and your interviewing experience!

When you're talking to companies and recruiters at career fairs and hackathons and over the phone, be confident, not arrogant. Be grateful for the opportunity they're giving you, and smile! A great attitude will take you VERY far.

## Your Skills

Obviously, your skills are what a company is buying from you when they hire you. So, honing in those skills and presenting them in an effective way is *probably* the most important thing in getting a gig.

---

## Building Them

Chances are, if you've had a data structures and/or algorithms class, you're already going to do pretty well in the technical interviews. That is, if you can recall the information you learned.

Here's a checklist of things that you should probably be prepared to know in a technical interview:

- Data types
- Basic Bitwise Operations
- String Operations
- Arrays
- Linked Lists
  - Singly Linked
  - Doubly Linked
  - Circular Linked
- Queues
- Stacks
- Heaps
- Trees
  - Binary Trees
  - Binary Search Trees
  - Tries
  - Self Balancing Trees
- Traversing Trees
  - Breadth First Search - BFS
  - Depth First Search - DFS
  - Preorder, Inorder, Postorder
- Graphs
  - Dijkstra's Algorithm / A\* Search
- Hash Maps
  - Handling Collisions
- Sorting algorithms
  - Insertion
  - Selection

- 
- Merge
  - Quick
  - Time Complexities
  - Programming Paradigm
    - Dynamic Programming
    - Object Oriented Programming
    - Asynchronous Programming
    - Functional Programming

This guide isn't for teaching you these skills. But there are several guides, problem sets, companies, and practice systems out there that can help.

### **General Guides**

- Sorting Algorithms
- Big-O Cheat Sheet
- Top 10 Algorithms for coding interviews
- Time Complexity Analysis in Data Structure
- Technical Onsite Interview Survival Guide

### **Problem Sets**

- Data Structures Questions
- Google CodeJam Practice Questions
- InterviewBit
- Tech Interview Handbook - Algorithms

### **Online Judging Systems**

- LeetCode Online Judge
- HackerRank Online Judge
- Project Euler
- InterviewBit

### **Mock Interviews**

- Interviewing.io - free
- Pramp - free
- Careercup - paid
- Gainlo - paid
- Impact Interview - paid

- 
- Meetapro - paid

Here's some books that might also be useful.

- Algorithms, 4th edition, by Robert Sedgewick and Kevin Wayne
- Introduction to Algorithms, 4th Edition, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (also referred as CLRS)
- Think Complexity, by Allen B. Downey
- Cracking the Coding Interview, by Gayle Laakmann McDowell
- A Philosophy of Software Design, by John Ousterhout

Typically, for an internship or your first job, questions won't get much more specific unless you're applying for a specific role. For example, if I wanted to be a mobile intern, I might be asked iOS/Android-specific questions, or if I wanted to be a web intern, I might be asked HTML/CSS/JavaScript questions, or if I wanted to be a backend intern, I might be asked about Django or Node or Ruby on Rails. That definitely depends on the company, and what you're both looking for.

Make sure you're able to implement and use the above data structures without the assistance of an IDE. Typically, onsite technical interviews involve coding on paper or a whiteboard, and phone/video interviews usually involve some kind of collaborative text editor. Yes, that means you won't have access to auto-complete, auto-compiling, or the internet to check for errors. So be ready for that!

## Selling Them

When you're actively emailing and speaking with recruiters, they're going to ask you a lot of questions that are just checkboxes for what they're looking for in terms of skills.

If you've written anything in a language before, **put it on your resume**. A lot of companies have parsers that look for specific things that, again, will put a tick on those checkboxes before putting you through to the next round of review. Now, if you've only ever done "Hello, world!" in Python, don't say that you're a Python ninja or whatever. You don't want to be thrown into an interview that will only hurt your self-confidence. Speaking from experience. Trust me.

When a recruiter or engineer is asking you about a certain project you've done, or how you've used a particular language, be as specific as possible. Tell them **exactly** what you did on that particular project (or internship or what have you). Tell them how much you contributed, what languages you used, how long was the duration of development for that project, what was the outcome of the project, etc. etc.

For example, don't say, "I'm the webmaster for a club website. Next question." Okay, Dwight Schrute. Go back to your beet farm. Instead, say something more like this: "I developed and currently maintain the website for my university's computer science club. I built it from the ground up with HTML, CSS,

---

JavaScript, and jQuery on the front-end. It's a static site, so a backend wasn't needed. The website's main function is to promote the club and give members crucial updates about meetings and events, and I update it regularly via a Python script I wrote." Oh my, you enchanting software engineer, you. Let me hire you.

When you're talking to companies about specific things you've done, make sure they know:

- **What?** - What? - What did you make? What does it do? What impact has it made? What was the hardest part? What could you have done better?
- **Why?** - Why did you make it? Was it for a hackathon, a school project, an open source contribution, or something else?
- **How?** - With which technologies did you make this? Did you use a specific API? What parts of it did you work on?
- **When?** - Did you do this recently or are you still living off of when you peaked in 10th grade?
- **Who?** - Did you work on this with anyone? Who did what? Who is this for?

Your skills are something that you have, and the reason why you have them is because you've used them in some way. All you have to do to prove yourself is to explain yourself! No need to go overboard and brag. Just be honest, and confident.

## Your Search

You have a resume and cover letter in hand, and you have the skills to make anyone want you. Now, you just have to find the right gig for you.

## Events

When you're on the hunt for a great internship or first job, the events you go to can really set you apart, and will help you meet people that could potentially help you in the long run.

The biggies that you will definitely run into are:

- Hackathons
- Meetups
- Conferences
- Career Fairs

So, how do you find these events? They're happening all the time, you just need to know where to look. Firstly, ask people in the field. Talk to a mentor, a fellow student, a professor, a colleague... anyone could come through for you! I remember my first tech event I went to in college was because I



---

ran into a guy that I met orientation day. And now he and I are coworkers. Fancy that. Anyway, people can get you very far.

Otherwise, when people don't work, we always have the glorious internet leading the way. There's so many resources out there I definitely can't list them all. So I'll list just a few.

- Your School - Yeah, this is kind of an easy shot, but not a lot of people consider their schools as an event generator, when they should. Go on to your university's website and find the career fair, find a seminar, find a company presentation, find something that will help you learn and meet people. You'll probably find a lot of options!
- Meetup - Meetup is a great place to find groups of people who are interested in the same things you are, who meet regularly. Look for engineering meetups that are hosted **at** companies. You'll get to see an office, and meet people who actually work there (in addition to those who might work at other organizations).
- Eventbrite - Eventbrite is best for finding events that aren't recurring. You can often find workshops, networking events, parties, and classes there. I tend to just search for the term, "tech" or "computer" and the results simply flow. Like a babbling brook. Or something.
- Major League Hacking - MLH provides a fairly thorough schedule of university hackathons every semester. Most of the events on their schedule provide travel reimbursement, but there's so many that you'll likely find one near you!

Like I mentioned before, there's a lot of resources out there for finding events. Now that you have places to go to, what do you do there? WHAT A CLEVER SEGUE.

## Networking

The whole point of going to events (besides learning something, of course) is to network. Meeting and maintaining relationships with people in this industry is essential.

So, how **do** you just network with people? How do you make it as natural as eating pie on Pi Day? I'll show you, Bert.

First of all, don't think of it as networking. You don't want to get the heebie-jeebies. Think of it as meeting people who like what you like. When you're at these events, you'll occasionally see groups of people staying in their own groups, not often reaching out to others. You're not going to do that. You're going to be a professional, social butterfly.

Follow generally these steps (as casually or formally as the event calls for) when networking:

1. Introduce yourself. This seems obvious, but you'd be surprised how often people just sort of slide into a conversation without ever actually providing anyone with their name or title. Go up to people and tell them who you are.

- 
2. Make small talk about tech. Again, an obvious tip, but important. You want to know what this person's skills are, and you want them to know yours. Someday, they might be looking for someone to help them on a project in a language that you know, or vice-versa. Get the important details out in the open about each other so that not only you can remember each other, but you can help each other out in the future. And don't forget to have fun with it!
  3. Exchange contact information. If you have no way of finding the person you're talking to again, and they don't have a way of finding you, then you're just wasting a potentially valuable connection. Whether you give them a business card or an email address, or even a Twitter handle, keep that information!
  4. Follow-up. This is probably **the most important thing I will tell you in this whole guide**. Hence the bold letters. So pretty. Anyway. Following up is the guaranteed way of leaving an impression with someone. Whether you email or tweet or InMail or Facebook message or pigeon mail, just sending a simple, "it was nice to meet you!" message is absolutely **vital** when it comes to networking. I've actually heard stories of people changing their mind about hiring someone based on a follow-up message. And following up doesn't stop at the "nice to meet you" message. At some point down the line, it's great practice to send a "how are you doing?" message to someone you're interested in maintaining a relationship with. For all three of my mentors that I had in college (and to this day), the way we built our relationship was through these sorts of follow-up messages. **Do them.**

Networking is about building relationships, and what you can do with those relationships is up to you. But we're talking about getting a job. So let's move on to one of the best ways of getting one, with these relationships.

## Referrals

Referrals are your in. If you can get a referral from an employee of the company you're applying for, that's money. It pushes you towards the top of the pile of resumes that they're getting every single day. Using the relationships you've built from networking, all it takes is a simple ask! You can get a referral from current employees, former ones, interns, former interns, friends of engineers, acquaintances of recruiters... really, anyone who is connected with the company. If they have a job posting up for a given position, they are looking around for engineers just as much as you're looking around for a great gig! The company knows that they're going to be speaking with someone potentially really good (because they were recommended by someone trusted), and you get to speak to a company for whom you want to work. Having a referral is a win for both parties.

Asking for a referral really depends on your relationship with the referrer. If you know them in a strictly business setting, you might want to send them a more professional request. But if you know the re-

---

ferrer pretty well, chances are you know how to ask for a favor. One thing that you must remember though is to, again, **follow up**! If you haven't heard back from your referrer, reach out and check on your status. If they let you know that you're in the system, be gracious and take them for coffee or something. Common courtesy. You've got this.

## Cold Calling

A lot of people are fans of cold calling, which is making unsolicited contact with a person at a company and hoping for something good to come of it. I admit, I'm not one of those people. Seems spammy. But, I'll tell you about some ways to do that anyway, to make your cold calls as potentially successful as possible.

First of all, you should have an email for someone at the company you're looking at. You could ask around your network for contact information for a given person, or you could just go out on a limb and test some emails to see if they're valid. A lot of companies have fairly standard email address templates (`firstname.lastname@company.com`, for example), so you can just keep sending a few until you find a legitimate one. If you need names to test out, you can scour LinkedIn or the company's website to try and find something that works. There are also many free tools to find someone's email address, given their LinkedIn profile.

Once you have a legitimate email on your hands, get to work on making a personalized message to them. Emphasis on the personal. When you're cold calling, you can't have just any standard email template where you stick in the company's name where it fits. You have to clearly let the person know why you're writing, from where you got their information (if it's not just a random guess, of course), why you would be a good fit for the company (where you will **sell** those awesome skills of yours), and why you think the company is great. Essentially, you're writing a glorified cover letter.

When the email has been sent, you're done. If you haven't heard back in a week or two, send a follow-up, but if they still haven't replied after that, chances are they won't reply at all. And that's okay. It happens. That's the territory when it comes to cold calling.

## Fellowships

Fellowships are a great way to network with a community and get valuable mentorship that will most certainly help you in the long run. They vary depending on the program, but typically you'll get assigned a mentor or two, go to several events to network with professionals, and intern for a company that has partnered with the specific fellowship program.

Here's a list of some example fellowship programs. This is by no means a complete list (feel free to send a pull request or file an issue if there's another you'd like to be added), but it should get you

---

started!

- Cansbridge Fellowship
- CODE2040
- Tech + Society Fellowship
- hackNY Fellows
- 8VC Fellows
- Mayfield Fellows Program
- True Entrepreneur Corps
- Venture for America
- Conservation X Labs Epic Fellowships
- Pursuit Fellowship
- MLH Fellowship
- Insight Fellowship

In addition to these fellowships, several companies offer special programs for younger students that are similarly geared towards learning and mentorship. Here's another list that is not complete, but will get you started:

- Microsoft Explore (I'm biased because I did this one, but this program is AWESOME)
- Google STEP (Student Training in Engineering Program)

## Conclusion

So, cliché parting words. Work hard, stay healthy, and be yourself! All of these phrases are said a lot for a reason.

You are in an incredibly challenging field that forces you to think in ways that many can't even fathom. You're constantly pushing the limits of logic and creativity, as well as technology itself. You've made SUCH an exciting choice to pursue this! It will be hard. You're going to want to give up at times. I certainly did. But, it'll all be worth it in the end when you've finally **earned** the gig of your dreams.

When I tell you to stay healthy, it's something that you might not consider when you're working until 3AM on a programming problem or trying to survive with 21 credits of technical classes. We've all been there. At times, it feels like it's the only way to get ahead. But guess what: it's not! Try to have a decent sleep schedule (as in, more than 3 hours a night) and eating schedule, too. It'll keep you from burning out, and it's proven that it'll keep you alert and focused, which is essential for interview season!

My favorite quote is from Helen Keller: "One can never consent to creep when one feels an impulse to soar." Everyone wants to be great, in their own way. When I say to "be yourself," I mean that I want

---

you to show people who you are, what you're passionate about, and what you strive to be! If you keep your personal goals in mind as you work towards them, everything will fall into place, and you can truly **soar** into a great career.

Alright, I've said my piece. Go get 'em. I hope that you find this guide useful as you go out to create change and be the best that you can be!

**Good luck!**