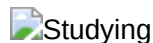

title: "Applying for a Software Internship remotely." date: "2019-12-06"
description: "I was in China studying abroad when I applied for internships. In this post I document my experiences of the process, as well as showing you how you can get an internship too." type: "blog"



The reason I'm making this post is that when I was applying for internships, I read many posts about applying, but none of them properly covered anything that I experienced, so here's my account of applying for internships.

I was studying abroad in China when I started applying for internships. Therefore I didn't have opportunities to do in person interviews as I was mainly applying for internships in the UK.

At my university (the University of Nottingham), students get a ton of help and support regarding getting an internship.

However since I was in China, I didn't get any help, except for a list of job openings in the university directory. This meant I did everything without the help of my university.

First, I'll be honest: **It's tough.**

Some companies won't even go past the CV stage if you mention that you can't go for an in person interview.

While you are at a disadvantage by applying and interviewing completely remotely, it's definitely possible with some preparation.

This post will be split into x stages, from preparing your CV all the way to online interviews. Start from the stage you're currently at.

Note - This post assumes you already have created projects, and know the basics either from self study or from university. Here is a great [link](#) for helping guide you if you haven't already done that, come back afterwards.

Stage 0: Learn the fundamentals

This might seem a little early, but I assure you it's not. You want to be consistently learning the fundamentals as early as possible. Cramming has never personally worked for me, and you will have more confidence going forward if you start at this stage rather than after you start applying.

Here is my (non-exhaustive) list of *fundamentals*, things you must absolutely know.

Here is the list below in no particular order:

- Trees
- Graphs
- Linked lists
- Stacks/Queues
- Arrays/Lists
- Hashtables/Hashmaps

- Sorting algorithms
- Threads/processes (parallelism)
- OOP stuff

For each item on the list, you must be able to code, explain and justify why you would use it, giving positives and negatives.

For algorithms, you must know time and space complexities, but also understand *what that actually means*.

If you were like me, you would have already encountered most things at university. However you don't necessarily **understand** these topics in depth enough to code them from scratch, etc without the help of google.

Even if your internship isn't completely related to OOP, you may still get asked questions about OOP like inheritance, abstraction etc. This was the case for me in several interviews.

Also note that although you could use things like flashcards to simply remember space/time complexities, its absolutely not in your best interest to do so.

The reason is that you need to understand these algorithms, in order to apply them to different situations, as well as simultaneously be able to explain to interviewers why you chose to do so.

Once you get the basics down, next comes the holy bible of coding interviews: *Cracking The Coding Interview*.

For me this was the best resource I found for preparing. It really is a **godsend**. Since you're just an intern, focus on the easier examples, as you won't be expected to know the hard ones. Fundamentals are the most important.

Here is a great [link](#) for making the most out of practicing problems on [Leetcode](#), [Hackerrank](#), etc.

Stage 1: Preparing and sending the CV

Preparing your CV

The main things you need to put in your CV include:

Info Section

This section is a short, sweet summary of you, with all your contact details so readers can quickly find the most **important** information.

In my case, this included my:

- address.
- email.
- phone number.
- website
- github account
- brief summary of what i'm doing and what im looking for.

Here's what my summary looked like:

I am a year 2 UK university student studying abroad in China for a year. I am currently looking for internship/year in industry opportunities.

Notice how it's brief and straight to the point.

Education

Simply write where and what you are studying, for example:

Bsc Computer Science - University of Nottingham

Also include how long you have studied there e.g:

September 2017 - September 2019

If you have good grades you can also note them down, if not, well.. *dont*.

Don't list out *all* your grades though, just put your average, as well as perhaps your predicted grades.

Experience

Try to keep the experience section completely related to software development. For example, I was lucky enough to go to a software company my uncle worked at to do a weeks of work experience. I also helped out at a Game Studio, doing black-box testing so I also noted that down too.

It's not usually worth putting any unrelated experience e.g bartender, unless you have nothing related to software development to put on your CV.

They aren't going to expect you to have heaps of experience. But anything that shows passion toward the field is a great plus.

When describing what you did, try to show how you **provided value**, for example:

achieved a reduction in time wastage by automating basic tests for the front-end.

(If you can give metrics, that's even better.)

However, a word of caution. Don't exaggerate too much.

Whatever you put in your CV is fair game to be asked about in interviews. Expect questions such as "how exactly did you provide value?"

Be prepared to answer that.

Personal Projects

This is where you can really shine, they might not expect you to have much experience, but they will want you to have demonstrable interest.

This is exactly the place where you can show that.

Don't go crazy however. I listed 3 personal projects, which would change depending on the company I was applying to. For example if it was a financial company, I would rotate in my project to do with cryptocurrency.

For each project, I wrote a simple summary of what it does, as well as the technologies used to build it. You don't want to go too technical here, leave that for the interview.

(Once again, make sure you are prepared to answer any questions regarding your projects).

I also mentioned the main challenges I had with some of the projects, e.g for my personal website, I had an issue storing image files as BLOBS within PostgreSQL, which "showed me how to correctly manipulate binary data when reading and writing to a file".

This is great bait for interviewers, who will most likely want to hear exactly how you solved the problem.

I also noted services I used such as docker, but never got questioned about them, presumably because the companies I interviewed for didn't use them.

Provide github links if your projects are open source. But make sure the code is clean for these projects, they won't expect perfection, but don't go full spaghetti either.

Skills section:

The skills section is an optional area to list other skills and hobbies you may have outside of programming.

For example I'm learning Chinese, so I put my chinese level, as well as my interest in tennis.

These can be great talking points to see if you match with the team when they interview you. Just don't fake it, put your genuine hobbies/skills.

Links

I didn't cover everything, so here's some useful links I used when creating my CV.

- https://www.reddit.com/r/cscareerquestions/comments/25u0eo/could_we_create_a_basic_undergrad_resume/chktg4y/
- <https://github.com/codebytere/so-you-want-an-internship>
- <https://www.vault.com/blogs/resumes-cover-letters/15-resume-tips-for-tech-jobs>
- [/r/cscareeradvice](#) do a CV check every Tuesday and Saturday. I posted mine and received some really sound feedback.

(extra tip - Make sure the CV is only 1 page maximum)

Sending your CV:

You've finished your first draft of your CV, time to start sending your CV out to companies.. **no, wait.**


First, send your CV to people you know and trust for honest feedback. It's important to get a broad range of opinions, so send it to non-tech people as well.

Listen to their feedback, and act upon it if it's a valid concern.

Once you have ironed out the first cranks in your CV, start to it send out to actual companies.

Keep track of what version of the CV you send to what company. This will help you figure out which versions are doing well, and which aren't.

Do not send early iterations of your CV to the companies you are most interested in, wait until it's refined and you see what works.

 Those are rookie numbers

Apply to as many companies as you can, and don't give up. It's tedious to do this, but with some thick skin it's absolutely worth it in the end.

Things like LinkedIn and Glassdoor are not your friends when you are applying as an intern. The reason for this is that these websites get so many applications that they are automatically filtered, which lowers your chance of getting an interview substantially without a lot of previous experience.

Cover Letters

I only wrote cover letters if they were required. The problem with them is they can take so much time to write, as you need to tailor them specifically to the company you are applying to.

Do not send out a generic cover letter, as companies will hire recruiters who work for other companies too. If they see the same generic cover letter, that's going to be an instant rejection.

Stage 3: Behavioral interview

So you've got a response from your application, and they want to arrange a phone call. Great news.

This phone call will most likely be with someone from HR. Their job is to make sure you're A) a human, and B) able to converse and backup your resume.

Tell me about yourself

The first thing they usually ask is the 'tell me about yourself' question.

Now for me, this was a really tough one because I wasn't sure how in depth I should go on this. Should I start from before university? Or keep it super brief so that they just know what I'm doing now?

The answer I found, is a balance of the two.

You want to keep it short, and start with what you're currently doing. I also then summarised what I did in my previous year at university, as well as the fact that I was studying in China.

Prepare for basic questions

It's common to not prepare for the behaviour interview.

But I think this is where a lot of people with great programming skills fall short.

Some common questions you will hear are:

- Give an example of your leadership.
- Give an example of overcoming a challenge.

- Give an example of failure.
- What is your greatest weakness?
- Where do you see yourself in 5 years?

In order to do well, it's much easier to get some **'stories' prepared** before the interview. If you're creative enough, you only need to prepare around 3 different examples, that will fit most questions they can throw at you.

In my case, I prepared a story of both leadership, and overcoming a challenge:

Story Example

(first state the 'problem' that you will be solving)

Last year, I had a group coursework where I was the leader. Two members had many arguments and therefore refused to do their work properly.

(next summarise how you resolved the issue)

To resolve this, I talked to each member individually to understand their issues with each other. I then emphasized that they are invaluable to the team and we can't complete the work without them. This led to a compromise between the two members. We also set up a trello board to properly track progress, as well as increasing meetings from once a week to twice a week.

(finally show how this provided value)

This resulted in us meeting deadlines, as well as getting a good grade in the project.

This is a simple example, but it's general enough to cover lots of questions.

I'd also like to note that it's a **true** story, and I advise you to use real examples as well, not ones that are artificial.

Research the company

This is something that is often overlooked for some reason. Whoever is interviewing you will want to see real passion and interest at the prospect of you working for that company.

Therefore, researching the company so you know exactly what they do, is essential and often interviewers have asked me what I know about the company.

Candidates who don't research the company might get caught out by such questions, being branded as lazy for not researching the company.

The 'pre-coding' coding interview

Sometimes, in order to filter out candidates due to so many people passing behavioral interviews, they will give you a 'pre' coding interview.

In my experience, this consisted of basic questions, often related to OOP concepts. Not actual coding challenges, but more general knowledge questions that you would be expected to know from your studies.

They will also ask you questions based on your CV, specifically about your experience and personal project sections.

Final notes

Ensure they know you will be interviewing remotely, unless they offer to pay for your travel expenses.

Thank the interviewer for their time. Throughout the entire interview, make sure you are intently listening, as well as being enthusiastic when talking about things related to the job.

When arranging interview times, account for time difference. I was 8 hours ahead of the UK, so I had to specifically ask if I could interview in the morning UK time, as that it was early enough for me.

In every behavioral interview, I was asked about why I went to China. I think one reason is that they were genuinely curious, but it also allowed me to give reasons such as I wanted to be more flexible and get out of my comfort zone. Therefore if you've done something similar, definitely mention it.

Stage 4: The technical interview

This will definitely be the main challenge, but presuming that you have prepared in advance, it's not as bad as you think.

An overview

This technical interview should last around an hour, possibly more.

The first thing that happens as you might expect, is introductions. It's not uncommon for multiple people to be interviewing you, and in some cases I had up to 8 people in a room interviewing me.

Next, I was drilled about my CV. So be prepared to talk about anything on your CV, for example design decisions within your personal projects.

After these technical questions regarding your CV, will be the 'whiteboard' challenges.

However since I was interviewing remotely, I had to record my screen via software such as skype, or something similar.

They also ensured any linting etc would be turned off.

For the coding challenges, I decided to use python, as I knew it well, and it's easy to turn pseudo code into python code.

Often the interviewers would guide me if I was going down the wrong path, or if I was completely stuck. They want you to succeed, their main job is to see that you can reason well, and solve problems.

In my experience, interviewers won't expect you to know everything when giving you problems to solve. Especially at non FANG companies.

I was often complimented on how clear I was in how I was trying to solve problems, and I think that's what mainly got me offers.

Advice for the coding challenges

As soon as they gave me the problem, I would immediately start speaking **ALL** my thoughts that I was having. I believe this is the **most important** thing you can do in your interview.

Articulating your thoughts is not an easy task, so if you can do it well, this is a huge advantage.

As well as clearly showing my thoughts, before writing code I would always try to visualize the problem.

Actually, for all the interviews I had, I used a whiteboard and showed it to interviewers via webcam. This not only showed them what I was thinking, but also helped me understand the problem more easily.

Next I would usually go into some super basic pseudo code. Once I thought it seemed correct, I would then dive into actual coding, making comments as necessary.

Furthermore, try to ensure you keep variable names tidy. It's really important to make your code readable. Since this is an interview, this will be their impression of how you code.

Usually, the questions I got were relatively basic, such as generating fibonacci numbers.

When going into interviews, it can be extremely nerve wracking. But for me, it was the same kind of nerves as taking an exam, which I had already learned to handle.

Therefore before interviews I would always say to myself that I had done the preparation I could do, so all I could do is give it my best shot.

How to prepare for this

As previously mentioned, Cracking the Coding Interview is great. But it's not enough. You need to actually practice these types of questions, with mock interviewers.

I did this with my room-mates several times. The first time I did a mock interview, I absolutely bombed it, as I was nervous, even though it was just my room mate.

However after doing it several times over, I started to get more confident, and after lots of feedback, was ready for actual interviews.

This is extremely important, you should find others who are trying to get internships and take turns. If it's online or in person, it doesn't matter. But practice is extremely necessary for this.

Invest in equipment

Make sure you are responding to questions confidently and clearly.

One of the best investments you can get towards this is a decent microphone, as well as speakers. No interviewer wants to hear your terrible microphone, and it *could* affect their decision to hire you or not.

Stage 5: (Hopefully) Getting an offer

I was surprised to get offers back so quickly from some of the companies I interviewed for.

I didn't accept any immediately, and neither should you.

Wait to see if you get more if you are expecting responses, but don't wait too long. Pick the one you feel works best for you.

Conclusion

Finding the first job/internship seems impossible because it's hard to look good without any experience at all. Add on the fact that you're applying completely remotely and it's even harder than for most people.

Once you find that first internship, however, the experience rewarded is worth all the struggles beforehand.

Good luck to everyone.