

CS Intro students Q&A

Presented by Oxy Open Source Club

About Oxy Open Source

1. [How do I get involved with the Open Source Club?](#)
2. [What programming languages are the most popular in the club?](#)
3. [Do you work on projects in the club?](#)
4. [Are there appropriate activities for people that are new to coding?](#)

About Majors/3-2 Program

5. [How feasible is it to take on Computer Science as a second major as a junior \(assuming most of my requirements for my first major are almost completed\)? If I am a junior, can I still minor in CS?](#)
6. [Is double majoring in MAC and CS a good idea?](#)
7. [Is there anyone doing a double major in CS and Bio/Biochem? Would like to get in contact with students who are double majoring in CS.](#)
8. [How does CS comps work?](#)
9. [Who can I talk to about the guidance for 3-2 Engineering Program for CS?](#)
10. [Is getting a B.A. in CS the same as getting a B.S. If not, what is the difference? How will getting a B.A. in CS impact me as an applicant for a career, versus getting a B.S?](#)

Course Related

11. [What is the best way to study for a CS exam?](#)
12. [After taking intro to CS, how proficient will I be?](#)
13. [I am having a hard time making friends in the class. How do you recommend forming study groups?](#)
14. [For someone who has not taken many math courses, should I be concerned? How much math should I know?](#)

15. [What are the biggest unexpected challenges when taking a CS course or majoring in CS?](#)
16. [What resources can I use if I feel like I am struggling in CS? Who can I turn to?](#)
17. [Should I be worried if it is early in the semester, and I am already struggling to code and/or understanding the concepts? \(I feel like I'm not knowledgeable in tech\)](#)
18. [What are some tips for getting through Data Structures Homework, or comp sci HW in general?](#)

[About Self Growth](#)

19. [Recommendations for learning programming languages on your own?](#)
20. [I want to start working on projects to add to my portfolio. How can I do so when I don't know where to start?](#)
21. [I am in data structures now, but want to do well in technical coding interviews. Will I have enough experience to be successful in a coding interview if I only have two months of Data Structures knowledge? If not, then what can I do?](#)

[Miscellaneous Topics](#)

22. [I don't understand how Github works.](#)
23. [I want to make JRPG.](#)
24. [How do I add SSL certificates to a webpage?](#)
25. [How much of CS is about computer hardware and history as opposed to programming?](#)
26. [People say the CS hype in terms of employment is dying, is this true?](#)

About Oxy Open Source

1. How do I get involved with the Open Source Club?

First, sign up for the [mailing-list](#) and join our [Slack](#)! Then, come to our workspace/workshops/events, and join our projects.

2. What programming languages are the most popular in the club?

- I use python, java, JavaScript, C#, php, HTML, CSS, SQL
- I know some Java, Python, Assembly, C, Abel, Haskell, Markdown, Latex. I think the most common ones in the club are definitely Java and Python.
- Python! (Python, Javascript, Java, C, C++ are the popular ones generally.)

3. Do you work on projects in the club?

Yes! Talk to Ruth (rschlosser@oxy.edu) about project ideas or participate in one of the existing projects. Soon, current projects will be listed on [our website](#) as well -- stay tuned!

4. Are there appropriate activities for people that are new to coding?

Yes! Almost all of our workshops are introductory level. And we host a weekly [Workspace](#) every Sunday 1-3 pm PST and Wednesday 5:30-7:30 pm PST. You can come here to seek help from SSAP tutors for homework questions, or practice intro, intermediate, or hard LeetCode problems to get ready for technical interviews. We also provide free snacks during workspace (when school is in-person)!

About Majors/3-2 Program

5. How feasible is it to take on Computer Science as a second major as a junior (assuming most of my requirements for my first major are almost completed)? If I am a junior, can I still minor in CS?

It is almost impossible to major in CS from scratch starting junior year because all upper division classes require CS229 as a prerequisite. Unless you have taken CS131 before, you would have a chance in finishing the major but your schedule would be packed. Yes, minoring CS is a lot easier.

6. Is double majoring in MAC and CS a good idea?

As long as you love both majors, it is a good idea! Double majoring is time-consuming, but also very rewarding. Be sure to plan ahead for major requirements in order to ensure that you will be able to complete everything. Alternatively, you could do CS+X, X being your second major of interest.

7. Is there anyone doing a double major in CS and Bio/Biochem?
Would like to get in contact with students who are double majoring in CS.

We don't know of anyone double-majoring in CS and Biology in the class of 2020. However, previous students have double-majored in CS and Chemistry. You can also contact Sophia Yang (Class of 2021). She is considering CS and Biochem or CS and Chemistry.

You can also join our [Slack](#) space and ask in the #general channel, see if anyone shares similar goals with you.

8. How do CS comps work?

Talk to Profs if you are considering majoring in CS and want to know more about CS major requirements. You can also come to this year's comps showcase on Dec 4th. All senior students would come present their comps.

9. Who can I talk to about the guidance for 3-2 Engineering Program for CS?

Prof. Jeffery Miller (jeffreymiller@oxy.edu)

10. Is getting a B.A. in CS the same as getting a B.S. If not, what is the difference? How will getting a B.A. in CS impact me as an applicant for a career, versus getting a B.S?

The main difference between a BA and a BS is not the major itself, but the distributional requirements. At Oxy, you have to take a foreign language, take classes about US diversity, fine arts, etc. At colleges where you earn a BS, many more of your requirements are in math, physics, and engineering. In many cases, the major requirements are exactly the same between BA and BS programs.

In terms of jobs, it's unlikely that employers care that much about the degree. Even with a BA, nothing stops you from taking math and physics courses. It's much more important that you actually understand concepts, and can demonstrate them in the interview, than whether you got a BA or a BS.

Course Related

11. What is the best way to study for a CS exam?

From personal experience, understanding every lecture's material after class and understanding all the homework problems is the best way to prepare for exams. Doing so as a routine would make your revision job a lot easier before the exam. Understanding is more important than memorization for all CS exams I have taken.

12. After taking intro to CS, how proficient will I be?

In Prof. Justin Li's syllabus for intro to CS, he states that students coming out of the course should be able to:

- write small (if not always efficient) programs of your choice
- participate in computational research
- solve real-world problems using programming
- learn a new programming language after some experimentation

13. I am having a hard time making friends in the class. How do you recommend forming study groups?

If you have the opportunity to work with other people during class, whether it is discussion or doing an example problem, ask them for their phone number after and ask if you could study together sometime :) I met lots of people in the class who I otherwise would not know. Even if you don't have the chance to get to know each other a bit during class, try asking after class. You could also try asking for help on a topic you struggle with; most of our classmates are very friendly.

14. For someone who has not taken many math courses, should I be concerned? How much math should I know?

I took up to AP Calculus AB in high school and barely passed the class. I felt very nervous about taking math in college. Rather than take the computer science computational mathematics concentration, I retook calculus and also took Mathematical Foundations of CS. It helped me to take calculus over summer, so I only had one class to focus on. Professor Leonard also makes Mathematical Foundations really fun (at least I thought it was!). I do not think you should be concerned about your current math knowledge. You should take at least as much math as is required for your major/minor and more if you plan to have a career that requires lots of math or if you simply enjoy math.

15. What are the biggest unexpected challenges when taking a CS course or majoring in CS?

- Managing time and tasks would become hard at one point because you are dealing harder and harder problems, and sometimes you wouldn't be able to estimate how long it would take you.
- Perseverance is an essential skill to master. You will deal with a lot of frustration, but hopefully you will enjoy the "pain". :)

16. What resources can I use if I feel like I am struggling in CS? Who can I turn to?

TAs and professors are great resources who want to help you out and see you succeed. Attending office hours with questions on the topics I struggled in was incredibly helpful. Sometimes, I felt that I was so behind, I didn't even know what to ask

or where to start. I was nervous to ask questions in class because I thought that my questions were stupid or that I was holding the class back from learning the material. Chances are someone else will have the same question! I didn't realize that a handful of my other classmates felt just as confused as I did. Even so, reach out to your professor and let them know you need help. It's better not to wait right before an exam so you can practice/study the material.

The great thing about going to a small school is that you can get some more personalized help from TAs and professors. If you want to reach out to upperclassmen, many of them can help you too (and also shared the same difficulties previously). Also, you can come into the [Workspace](#) and ask questions to the students who are there as well.

17. Should I be worried if it is early in the semester, and I am already struggling to code and/or understanding the concepts? (I feel like I'm not knowledgeable in tech)

Do not worry! I often felt confused and started to feel like I was falling behind the class in 131. At around 6 weeks, I had a light bulb moment and the material started to make a lot more sense. Thinking computationally can be challenging, especially at first. It takes some getting used to! After time and practice, you will see improvement. Stay diligent :) Please talk to a TA or professor or a classmate and ask for help on the things that confuse you.

18. What are some tips for getting through Data Structures Homework, or comp sci HW in general?

Don't be ashamed of not knowing the answer. Like every other skill, practice helps you become more comfortable with both coding and theoretical concepts. Be persistent in finding a solution: there's Google, TA, professors, use these resources.

About Self Growth

19. Recommendations for learning programming languages on your own?

I have found Codecademy's free courses to be particularly helpful to learn new languages. I also highly recommend searching up Youtube tutorials for the programming language you are interested in and watching a whole playlist of them

together. The key is to follow along and write some code, even if it is just copying what is in the tutorial. Search up small projects on Youtube that use that language and build the project with them. After that, you can scan through the documentation and look through anything you don't know or are interested in. Getting to know the documentation of that language is also very useful knowledge. I have also found that checking books out of the library can be helpful; sometimes I need to check out multiple sources to actually understand what I am working on. It depends on the way you learn and your preferred kinds of media.

20. I want to start working on projects to add to my portfolio. How can I do so when I don't know where to start?

First, think of what kind of project you would like to work on and where your interests lie. Do you like video games, websites, web apps, phone apps, machine learning, etc.? There are a ton of Youtube tutorials for beginners to work on different kinds of projects. Also, you can try starting from a project someone has already worked on. You could check out popular repositories on GitHub and see if any of them interest you. If you want to build a website, you can fork someone else's repo and change it for your own use.

You could join Oxy Open Source and help us out with our projects. Our website especially could use some help. We have a few people working on the website already who can help teach you the coding languages and concepts: JavaScript, React.Js, HTML, and CSS.

21. I am in data structures now, but want to do well in technical coding interviews. Will I have enough experience to be successful in a coding interview if I only have two months of Data Structures knowledge? If not, then what can I do?

It depends on how well you understand what you're learning in class, but practicing more will always be better. Many people have recommended the book, "Cracking the Coding Interview" to prepare for interviews. On top of this, you can also practice on HackerRank and Leetcode.

Miscellaneous Topics

22. I don't understand how GitHub works.

There are a lot of great resources online, including GitHub's own [tutorial](#). OxyOS is also planning a GitHub tutorial workshop for this semester. Stay tuned!

23. I want to make a JRPG.

There might be a Game Design course next semester, ask Prof Leonard or Prof Celia Chen if you are interested. OxyOS's president Alexis has hosted a Game Dev workshop last year, you can also talk to her.

24. How do I add SSL certificates to a webpage?

Ask google. :)

25. How much of CS is about computer hardware and history as opposed to programming?

There are a lot more interesting things other than hardware, history, programming stuff in the Computer Science field. In terms of Oxy's CS course, programming classes and non-programming classes are roughly half half.

26. People say the CS hype in terms of employment is dying, is this true?

Not true. The need for CS talents in the job market still exceeds supply, especially with current immigrant policy, the gap is even larger nowadays.