

Advice to pre-350 self.

Here's a way to help students who will be taking 350 in the fall and earn some xp for yourself.

Knowing what you know now, what advice would you give your past self a month before taking the class. Or, put another way, what advice would you give students about to take this class?

If you have been stuck on one aspect of your assignment for a while seek help. There are many small things to get hung up on in every project of this course that can suck up a lot of time.

Read the chapters from the book before each RAT so everything is fresh in your head. Also, don't put assignments off because there is no due date. You don't want to have to complete multiple assignments under the wire.

Get at least a basic understanding of HTML before you start the course. It will take some extra time in the summer or over your winter break, but will definitely save you tons of time in the long run. Some of the rudiments took me a very long time to complete, not because I didn't understand the database portion, but because my HTML was all out of whack and I wasn't knowledgeable enough on it to work out those kinks. Definitely wish I had at least gone through a basic HTML tutorial or done something similar to brush up on my HTML skills before I began the course. Once you start classes, it's tough to find time, so do it before you start classes. It could have saved me many hours of frustration!

Set deadlines for yourself!! Even if this is something you're really excited to learn, you're going to fall behind if you give yourself too much freedom. Plan responsibly and work responsibly so you can learn more efficiently and be satisfied with the results. Also, when working with any web tools, it helps to get familiar with the developer tools in browsers. I was already familiar with them before this class, and they're incredibly helpful for debugging code.

OH and editing for one last thing: don't forget that you can speed up videos on Youtube!! If you're listening to one of the video lectures and you feel like it's not fast-paced enough for you, speed it up or slow it

down where you need it. The videos are still completely coherent when they're sped up, so you won't miss anything, and again, you can just slow it back down when you get confused.

This is a class that is meant to test you. You are supposed to learn a lot in this class, but you'll thank yourself later on. Those of you who already have SQL experience will start out faster than those who don't, but don't let that discourage you. There are really 3 main things you can do to make this class a lot easier for you if you are willing to take a commitment early on.

1. **Learn Python** - It is very important to learn a proficient amount of Python before you take this class. This class isn't meant to teach you Python; it is assumed that you already know some and/or you are willing to learn more. Before my Fall Semester, I knew I needed to know Python for Data Mining in the Fall and for Application of Databases in the Spring. So I took the initiative to learn Python fast. I bought [this book](#) to help me learn Python. I feel like Python is a very important language to know coming out with a computer science degree. There are other resources online, all you have to do is take the extra time out of your life to learn something new. That is what Computer Science is all about. You need to constantly keep learning and evolving; you can't be lazy when it comes to learning new programming languages.
2. **Try out SQL** - In this class we used PostgreSQL. It uses SQL. MySQL uses SQL. I heard enough excuses that "PostgreSQL is not the same as MySQL." So in my spare time, I wanted to challenge that claim. Out of all the code in the textbook and what I've found online, only 1 query did not work. So try to learn some SQL as well as Python because you will thank yourself later when you need to start getting into Joins and Subqueries.
3. **Use Bootstrap/Foundation** - The biggest help to me in this class was using a prebuilt frameworks for HTML. I used Bootstrap and it was really easy to learn how to make a decent looking website. This goes back to point 1 where you need to try new things. I used Bootstrap with my Twitter clone for CPSC240 and I fell in love with how beautiful websites become when you use it. If you take the time to learn it, you will be a commodity if you want to do some sort of web development.

One last piece of advice is to Google. There were so many questions I was asked where I did a simple Google search to find the answer. There will never be a workplace you will ever go to that doesn't allow Google. So learn how to *Properly search on Google*. If you do, you will be so much happier than if you waited around for someone else to help you. Don't be afraid to find resources that show you how something works. If you can look at the code or explanation and implement it properly, [you learned something new for the day](#).

This class is not going to be easy. You have to put a lot of hard work and time if you want to do well. If you start strong your time at the end will be a bit easier, but you need to remember to stay on top of your work even if you get ahead in the beginning. ASK QUESTIONS if you are confused. It is a hard hard HARD way to learn just by teaching yourself, so reach out when confused and that will make your life a lot better. Watch and then re-watch the videos. They go over a lot of important material so it is crucial to understand all of the important points.

Bookmark these links:

<https://flask-socketio.readthedocs.org/en/latest/>

<http://www.w3schools.com/sql/>

<http://www.postgresql.org/docs/8.2/static/index.html>

Learn some HTML, CSS, and JavaScript. Seriously. Go to Codecademy or something and take the courses on them and get reasonably comfortable using them and this course will be much easier. Even though most of the heavy lifting in this class will be in Python, almost every assignment will require extensive use of the the three previously-mentioned languages. Also, if you aren't already, get some experience with Python. This course is predominately in Python, but it's about utilizing databases with Python, not learning how to use Python. Fortunately, it should be pretty easy for a Comp Sci major to learn Python to a usable degree.

Also, if you have some experience with SQL, don't put things off because you already know how to do them. Do them as quickly as possible and you might be able to skip the midterm and get full xp for it.

At first this class portrays itself as a walk in the park and will be extremely easy, however this is not the case. Time management is more important in this course than your skills working with python and various other coding languages you will see this semester. Each rudiment will take you a great deal of time, so DO NOT wait to get them done. Believe me, I did this and am struggling to meet the deadlines especially with other classes. The biggest advice I can make to you is to make connections with your classmates, because they will be the biggest asset within this course. Don't be afraid to ask anyone for help because time is something you cant afford to lose in this class. I probably am making this course sound extremely bad, but in retrospect I have learned more in this class about coding, software development, and even my

work habits than any other CPSC course I have taken so far. Basically: don't procrastinate, ask for help when needed, make connections, and have fun with the projects

I would also stress that time management is key. While the flexible course schedule is meant to allow you to master a topic before moving on, don't languish over one project for too long. There is a lot of material to cover! If you are stuck, ask for help. Many times it's a simple issue that you just needed an extra set of eyes on. I fell into the trap of being proud and thinking I could figure it all out, which made me fall behind. Don't do that! It doesn't work! Ask early, and often!

Clear your schedule. Clear your schedule. And, oh yes, clear your schedule! You have a lot of work ahead of you. When an assignment should take approximately 10 hours, schedule 20 hours for it. Even though there are few deadlines, follow the instructor provided schedule to the letter. There are free-form opportunities, some in the individual assignments, but mostly in the group project. Ron provides opportunities to learn new things on your own, explore. Take advantage of it! Explore and learn new technologies. It is fun and educational. However, it takes time! Also, don't be so afraid of breaching the honor code. Ron awards Karma points for helping other students. Don't share code, but provide guidance and suggestions. And if you are helping students for hours on end after class, like I did a lot, tell Ron so he can give you points!

Take a light load for the semester because this class takes an absorbent amount of time. If you're stuck, seek help instead of spending 10 hours fixing a bug because you don't quite understand the error messages. Work hard and you'll get a good grade. Study a lot for the rat and lightning rounds as it can be the different of having an A in the class without doing too much of the nosql material and having to do all of it in order to get an A.

Also, hope you get into a good team.

First off, I would advise anyone who is interested in taking cpSC 350 to take a light course load that semester. Secondly, I would recommend that they brush up on their python (or learn it if they don't know any), HTML (seriously, study this), and postgresql. Having a decent foundation in these three subjects going into the class will allow the student to breeze past the first few rudiments, giving them a lot of time

to devote to the harder rudiments (socketIO part 1 and 2). Also I would advise the students to not get discouraged by a low XP score at the beginning of the semester. You do need to have a lot of XP to get a good grade in the class, but in the end, the students will find a way to get the XP.

Procrastination is the enemy! It may seem ok to put off working on some of the assignments because you have other homework due for other classes but dont! Get some coffee and a comfortable environment to code and plan on staying there a while because you will be there for a while. Brush up on your python, html, postgresql, css, etc. skills and make one big cheat sheet of common commands because that will save your programming life. Google is your best friend and ask for help even if you don't think that you will, always ask for help. Computer Science shouldn't be done all by yourself. You need people to help you debug if not bounce ideas/strategies off of if not for moral support. You can do it! It's not going to be easy but you will make it out in the end with some knowledge.

Take more time to work on projects and be very careful of simple errors. Try to find some extra time during the week to work on the projects so you can finish them quicker. Also, considering the amount of time wasted on simple five minute fixes (that then took longer than five minutes) be very careful with your syntax and making sure all your variables are of usable and correct types.

Write in your textbook and complete the exercises. The Head First book is *designed* to help you retain information, but only if you take the time to invest in this really great resource. A careful reading helped me immensely on the RATs and Lightning Rounds (an easy source of points if you read the book).

As far as projects go, it's a good pace to submit at least every two weeks or so. If a project is taking you more than two weeks, immediately reach out to Dr. Zacharski. On one project, I hit a wall and didn't even know what questions to ask, stagnating for weeks. I finally scheduled a private tutoring meeting with Zacharski, and I wished I had done so sooner. I finished the project within the week.
