edX

UBCx: SPD1x How to Code: Systematic Program Design - Part 1   **Help**

🔖
Bookmarks

▾ **0: Introduction**

Welcome

Course Organization

Syllabus

Grading

Study Tips

Course Team

Module Wrap-Up

▸ 1a: Beginning Student Language

▸ 1b: How to Design Functions

▸ 2: How to Design Data

▸ 3a: How to Design Worlds

▸ 3b: Compound Data

0: Introduction > Study Tips > To Do Well in this Course

To Do Well in this Course

🔖 Bookmark this page

Here are some tips for doing well in this course. We suggest you read this every week or so for the first few weeks of the course, that will help incorporate these ideas into your work.

1. **Program design is something you learn by doing** – if all you do is watch the videos and look at the solutions to lecture/practice problems, you will almost certainly fail this course. Instead you should:

   - Watch the videos actively: try having DrRacket open and typing along with the lecture. Just stop the video if you need a moment to catch up.

   - Use the pause button! If you are watching actively, typing along and so on you will need to pause the video periodically.

   - Try to re-work the lecture problems on your own from scratch after every lecture.

   - Do practice problems, working them from scratch before looking at the solution.

   In the words of a former student:

   > It's funny how repetition in actually doing the work leads to it all of a sudden making sense to me. I can't explain it. I don't have a sense of growing understanding. I have confusion, repetition, and suddenly a sense of understanding. It's not coming through watching or reading, though. It comes from replicating what I've watched several times. This has happened repeatedly for me in this class. I start the week with a sense of accomplishment, well being, peace of mind, and confidence from the victory over last week's homework. I am then submerged in doubt, confusion, and erm, terror. (Why not be melodramatic?) Suddenly there is a working epiphany and I have victory again.

2. Do not worry about making mistakes! If you make a mistake and get an error message read the error message, use it to identify the

✏️

problem and then correct it.

3. If you are unsure whether something works, try it! For example, the best way to answer a "does it work to do X" question is by trying it.

4. Take advantage of the discussion forums to ask questions.

5. Stay caught up, even on the early material which may seem slow/easy but actually isn't -- really internalizing it takes time and practice. This material also lays the foundation for the much more complex problems that follow.

6. When taking notes during a lecture video, focus on recording the design process, not the end solution (which is always available by pausing the video).

7. It is better to work a bit every day (e.g., 1 hour) than once a week for a large chunk of time (e.g., 5 hours).

8. Always be able to identify what step of the recipe you are on – if you are working on a design problem and you don't know how to proceed step back and ask yourself questions like: What step of the recipe am I at? What should I be doing at that step? Where should I be looking to figure out what to write? Try looking up a similar problem that you have already solved to see what you did at the corresponding step.

9. If you get stuck for more than 5 or 10 minutes on a design problem get help. Post to the forums and see if someone has a suggestion. Include **what step of the recipe you are at in your post!**

POWERED BY
OPENedX