CPEN 221 & CPEN 211 Review

Oct 16, 2020

Note: A very standard course review post, written almost half a year after having taken the classes myself. If this post were written any sooner, all you'd hear would be complaints!

CPEN 221

- Language: Java
- Tools: Intellij (Java IDE), Gradle (build tool), JUnit(Testing)
- Textbooks: N/A
- Professor: Sathish Gopalakrishna

My Rating: Workload: 4/5, Difficulty: 6/5, Usefulness: 5/5

Known as the follow-up course to APSC 160, this course goes into the meat of software development. It covers object-orientated programming, data structures, lambda functions, and some algorithms. Also involves in-depth software engineering principles including testing, specifications and error handling which are meant to help you write good code, as well as basic Java principles.

Basically, if APSC 160 is like learning algebra equations like y = mx + b in high school, then CPEN 221 felt like the entirety of calculus.

Breakdown:

1. Mini Projects

   Aka Massive projects (maybe I'm exaggerating). There are 3 in total, and you get to do them with a partner. There is usually some skeleton code and tests, but much more needs to be done includes getting the project to work. For example in my year, we had to create a Java app that stores audio as a frequency set, and play it back, add echos, etc, and another project was creating a server that sends requests to Wikipedia.

   These projects were scored out of 4, and I think it took me a few days to finish the work necessary for ¼. Get started early!

2. Weekly Exercises

   Leetcode style homework questions. You get 1/1 for each exercise as long as your code works, and gives the correct output. But don't underestimate these, I've taken as long as a few days to finish one, but definitely gets a lot easier as you learn more about Java.

3. Coding quizzes

   Similar to weekly exercises, but harder, and written in a timed-environment. You get ~75 minutes to code up a working solution, and this gets scored out of 4 as well. The problems are really difficult, in fact, a lot of people received 0's for the first quiz. I remember my friends and I all saying we were going to fail because one must pass this section of the class to pass the course. In the end, more sittings of the coding quiz was given for those that needed it.

4. Exams

   This is the theory-part of the course. Basically all the lecture material and readings are tested here. Doesn't involve writing any working code, but reading code and answering questions.

Material covered:

- Object-orientated Design: abstract data types & how they are realized through Class/Interface/Inheritance, and the 4 main oo concepts: Abstraction, Encapsulation, Inheritance, Polymorphism
- Testing (black-box, white-box, test-driven development, test sets)
- Mutability vs Immutability
- Recursion
- Equality
- Lambdas, Streams
- Exceptions & Error-handling
- Parallelism: Threads, Mutexes
- Data structures (not taught directly, but you learn them through doing exercises) including sets, arrays, trees, maps
- Algorithms like dijstra's & breadth, depth first search (also not directly, you learn them by doing exercises)

Course Experience:

First of all, this course is very intense, probably among the hardest courses I've taken up to date. It expects that you know Java right off the bat, as they throw you into weekly programming questions, Leetcode style. And as someone who didn't know any Java, learning to use the language, as well as trying to solve problems using the language in parallel was very difficult. Next, the material covered in this class is pretty advanced, considering most people's introduction to coding was APSC 160. And to top it all off, we get timed "coding quizzes", where we're expected to think on the spot and code a working solution to a multiple part problem in a 75 minute time interval on your laptop.

I absolutely hated this class when I was taking it. It seemed like the abstract lecture topics had nothing to do with the actual coding exercises we were supposed to

complete. Only retrospectively, I realized this was best and most useful course in software development out of all the computer engineering/science classes I've taken.

Not only do the lectures introduce integral software engineering concepts, the sheer amount of coding you do here will make you remember Java for the rest of your life. And of course, this class makes CPSC 221 and CSPS 261 seem like a breeze.

Recommendations for Learning:

Learn Java before taking this class. I was told this by an upper year student in CPEN, except I thought they had said "JavaScript". After a summer of learning JavaScript, I come to class and find out that Java is notJavaScript. In fact, they have nothing to do with each other.

Here is the video I used in the beginning to learn Java (really you can pick any video series).

However, this series is more beginner level, and you probably need a better understanding of some other topics. I picked up "Java: A Beginner's Guide" in the middle of the course, and it's this huge book, but covers a lot of the topics in question.

CPEN 211

- Language: Verilog (Hardware description language), ARM (assembly language)
- Tools: Quartus Prime (program that synthesizes hardware, downloads it onto the chip), Modelsim (simulator for development and debugging), DE1-Soc (board with buttons, lights and a chip)
- Textbooks: N/A
- Professor: Tor Aamodt

My Rating: Workload: 6/5, Difficulty: 4/5, Usefulness: 3/5

I don’t even know where to start with this course. Basically, this course uses Verilog, a hardware language. Code written with Verilog is then loaded onto a chip, which is inside of a board (DE1-SOC) and connected to various inputs and outputs like lights and buttons. Because this language is very “low-level”, it's generally harder to handle, and involves working with bits (1’s and 0’s) and wires (part of the hardware that is synthesized onto the chip). The latter portion of the course involves writing assembly code, which is another low level language. Why is this course so bad? The unreasonable amount of work (and super long lectures) for no reason at all. And the professor announcing every week that he caught students cheating, resulting in a feeling of hosility in the course.

Breakdown:

1. Labs

   This is the most work-intensive and aruduous part of the course, worth almost half the grade. Each week, there are labs done with a partner, which are basically coding homework assignments, and involve writing Verilog for the De1-SOC, which does some task in the end (remembers orders of switches, and lights up corresponding lights, etc). Nothing crazy, except that the lab assignment handout is 10-20 pages long, and the lab takes 10-20 hours to complete. Also, some labs build on top of prior ones, so don’t get lost.

   this video shows what an introductory lab would look like:

2. Lab Proficiency test

   These are timed exams, where you get a smaller problem that is similar to the lab, to complete in ~2 hours. Basically to get rid of all the people who rely on their partners to carry. As long as you actually do the lab, this isn’t too hard.

Material covered:

- Hardware things like state machines, flip-flops, multiplexers, arbiters, logic gates, boolean algebra...I really don't think it's useful of me to describe them. It's totally a different beast from normal programming lanugages.

Course Experience:

This class took up a lot of time, purely from the labs. I remember doing labs all Sunday, and then not finishing, and spending more time throughout the week to complete it. And then the cycle repeats for the next lab.

It's not necessarily hard, but arduous since it took so much time to complete. Also, lectures are 2 hours long, and they're not super helpful. Overall, difficulty is less than CPEN 221, but work load is more.

Recommendations for Learning:

Bite the bullet and work through the labs. Also, ask for help from lab TA's when needed. This course isn't super useful in the grand scheme of things for those who want to do software development (unless you would like to work for Intel and become a chip developer), but the 300-level class CPEN 311 builds on top of this material.

« More Posts