CPEN 3rd year, Part 1

A review of CPEN 3rd year

Unlike CPEN second year, third year is actually a much more relaxing ride unlike our friends in electrical engineering. It only gets better from here, and there is a substantial improvement in the workload of the courses. You get much more flexibility in choosing courses you like since you are not restricted by an STT, and you can also get your free electives done in third year.

Choosing courses

As a third year student, you are unlikely to be able to register for any advanced electives this year because they fill up quickly. I'd recommend getting your free electives (non ECE courses) done this year instead like I did.

CPEN 331 - Operating systems

- Language: C
- Tools: GDB, os-161.
- Textbooks: There are some readings but no textbook
- Professor: Alexandra Federova

Verdict

- Workload: 4.5/5
- Difficulty: 3/5
- Usefulness: 4/5

This course is, in my opinion, the successor to CPEN212. This operating systems course builds on many of the concepts taught in CPEN 212 that are briefly covered, and puts them into practice. This class is a lab-based class, where you will slowly implement elements of a "teaching" operating system known as OS161. Features you will implement include locks for concurrent operation, read and write operations, filesystem operations, and thread functions (fork and exec). For those of you who have taken CPEN 212, this will not be unfamiliar and is essentially applied CPEN 212.

There is a substantial amount of time commitment required for this course, and for the final assignments around 80 hours is required (or as suggested by the instructor) You can work in pairs for the last 3 assignments though, and you will be given sufficient time to complete the 5 assignments over the course of the term.

In my opinion, the lectures are not very useful, especially when my mind is preoccupied with trying to answer the clicker questions, which are correctness only. The slides are not very clear either, and I often found myself reviewing the 212 slides.

I also really hate OS161 since it is a maze to navigate through all the files.

Breakdown

1. Labs - 70%? The bread and butter of this course. You will be completing parts of the OS161 OS over the span of the term. The labs are autograded for functionality, and you will have access to the tests and parameters on how the tests will be run. This typically accounts for 60% of your grade. The tests test on functionality, but also whether your code is "safe". The remaining 40% is given for code style (and is basically free marks). They do take lots of time, so start early. In addition, it is also a pain to get OS161 working (maybe just for me because mac) because there are often a bunch of failures in the install script.

   You will notice there are lots of OS161 repos out there with completed implementations. As an advocate of work smart, not work hard, the code out there is a good reference, but only if you understand what it is doing. Copying it without understanding what is going on is a surefire way to get caught (the instructor knows that tons of solutions are lying around online). With the generous grading scheme in this course, it is often not worth it to copy. Use it as a tool for understanding.

   I'd also like to plug the website Pearls in life. This site was incredibly helpful in guiding your implementation. It basically tells you what you need to do, short of giving you the code.

2. Oral Exam - 20%
   For this part, you will basically show your code to the TA or the instructor and explain it, how you wrote it, and how you debugged it. Essentially, it is like a CPEN 211 demo. The oral exam is really chill as long as you know what you're doing, and you didn't copy the code. Some TAs are really clueless, so just speak really quickly and hope they get confused themselves. This is a pass/fail mark, so its basically free marks.

3. In class clickers - 10% Standard clicker questions. Not free marks and somewhat difficult. Some of them you can discuss, others you cannot. The clickers questions are on the slides, so you can go through them the night before and select your answers. I personally hate this teaching style since my mind is preoccupied with the clicker questions. Fortunately, some kind soul has put all the quiz on their github repo.

Course Experience A moderate course that takes time, but the marking scheme is really generous so it makes up for it. No midterms or finals is really nice as well. Find a good partner to do the labs and split up the work.

Recommendations

- Start your lab early

- Answer the clicker questions the night before. Do the pre-readings
- Don't use a mac. Maybe get a linux distro or something, plus a second monitor for all the code you'll have to read.

STAT 251 - Statistics requirement

- Tools: R / R studio
- Textbooks: Exists but not necessary
- Professor: Lasantha Premarathna

Verdict

- Workload: 1/5
- Difficulty: 2.5/5
- Usefulness: 1/5

Standard stats course, one of the many options you can take. This is the easy one, but if you want to take the CPEN machine learning classes, you'll have to do MATH 302 (or something like that.)

Goes into things like mean, median, mode, standard distribution, other probabilty distributions, analysis of variance, correlation coefficients, and other stat topics that I have completely forgotten.

Breakdown

1. Midterms / Finals - 22/45% A test of the content covered in class. The midterm was fairly reasonable, and if you understood the class content and examples, you could answer the questions in the time given. Final was also similar, but there was one unsolvable question. There are plenty of practice questions / past exams.
2. Webwork - 10% @ 1% x 10. Gives you good practice for content in class and for the exams. They are limited attempts which I hate, but the webworks have solutions someone has kindly posted on their github, and they are standard questions from which I think you can find the solution on the webwork github.
3. Labs - 8% @ 1% x 8, weekly Dedicated lab session weekly. You work in assigned groups to complete the lab within the session. Its nice since its basically just 1 hour of work. Graded pretty leniently. You get to practice R. The professor always threatens to put an R question on the exam, but I have never seen one.
4. Written assignments - 10%, (**5?**)% x 2 Two written assignments that are relatively simple. On distributions. Find a partner to verify your answers to ensure no silly mistakes since it is straightforward to complete.
5. Clickers - 5% Happen every class. 1 point for participation, 1 point for correctness. You can discuss with people around you. Moderate difficulty, but he does give freebies close to the end of term when no one shows up.
6. Bonus - 1% for completing the survey
   This one is a scam. He claims to give out 1% extra if 70% of people finish

the teaching survey, but I think he says this because he knows that it is basically impossible to get 70% of the class to do something. He even gives in class time. Let me know if he actually does give out that 1%.

Course experience A well taught class that covers everything you need to know in the lectures. Relatively straightforward degree requirement, but pretty useless for CPEN students. I have forgotten everything already. The assignments and exams are reasonable and he is responsive to student questions and feedback. Overall a pretty good course.

Recommendations Find a friend to verify your answers for assignments and clickers. Do the practice questions. Otherwise pretty straightforward.

CPEN 322 - Web applications

- Languages: Javascript
- Tools: Node.JS, mongoDB
- Textbooks: none

Verdict

- Workload: 3/5
- Difficulty: 3/5
- Usefulness: 4/5

A basic introduction to web apps and technologies behind them. You will be introduced to the frameworks behind which websites and webapps operate, and fundamental concepts that will be useful. These include asynchronous operation, promises, characteristics of object oriented programming in javascript, closures, event handling, DOM manipulation, AJAX, files, streams, databases, REST API, and some basic cybersecurity.

Breakdown

- Assignments: 40% @ 8% x 1
  Over the course of 5 assignments, you will create a web app with more and more features. They are autograded and tests are provided to you. Other than checking for hardcoding, there is no manual grading. They are relatively straightforward and easy. They give you experience implementing theoretical concepts into something tangible.
- Programming proficiency test: 5%
  Just basic for loops and programming to make sure you are familiar with javascript. Pretty straightforward, unlike CPEN 221.
- Weekly quizzes: 12% @ 1% You will watch ~2 hours of videos covering the content you will need to know. After that, you will complete a canvas quiz on the videos. Medium difficulty, and you need to read carefully. I personally hate these timed quizzes but fortunately someone has posted solutions on github.

- Midterm / Exam 12/25% CPEN 221 style leetcode questions on concepts covered in class. There is insufficient time to do the questions considering the difficulty, and most people just get 50% of the questions. Autograded but test cases provided.

Course Experience Personally I hate flipped lecture classes because they burn twice as much time as normal classses since you have to watch videos AND attend lectures, which are basically where he goes over examples. The exams are also incredibly difficult to make up for the free marks for the assignments.

Overall a reasonable class.

Recommendations

- Actively participate and get your Node JS running in your console to try the in class examples. This will help your learning.
- Try using typescript (if you have time) and transpiling to JS for your assignments. The type checking will reduce errors and debug time. Dynamic typing is the worst for creating all sorts of weird bugs.
- The instructor has a really flexible policy of replacing any of the assignment / midterm marks if you do not attend. The weight will be transferred to the final accordingly, but this is basically a triple edged sword since the assignments are free marks, while the final is incredibly difficult. Don't do it.

CPSC 320 - Intermediate algorithms

- Languages: Pseudocode
- Tools: Pen and paper, Latex
- Textbooks: Algorithm Design (Jon Kleinberg, Eva Tardos)

Verdict:

- Workload: 4.5/5
- Difficulty: 4/5
- Usefulness: 5/5

Breakdown

Assignments (5)	30%
Take-home Tests (2)	20%
Midterm	15%
Final Exam	30%
Pre-class Quizzes	3%
Tutorial Quizzes	2%

To be finished.