

CPSC 315

Programming Studio

Yoonsuck Choe

People

- Professor: Yoonsuck Choe
- Teaching Assistant
 - Randall Reams
- Peer Teacher
 - Cody Taylor

Background on the Course

- Meant to be a “capstone” to the lower-level classes.
- Intention is to give **lots** of programming experience, in a team environment.
- Should be prepared for any programming assignment in upper-level classes
- Should be better prepared for industry programming jobs (internships/co-ops)

“Studio” Course

- Programming as “art,” “science,” “engineering.”
- The idea of a studio course is to have an environment where students can practice and refine their skills
 - Your skills should markedly improve over the semester
 - You should have plenty of interaction with and feedback from the professor/TA/PT
 - Practice, practice, practice

Lectures

- We'll meet a minimum of 2/3 of assigned lecture periods.
 - Expect to meet most dates at the beginning of the semester
 - Will skip lectures later in the semester and during projects
- Lectures should be helpful for your programming work

Code Construction: Where It Sits (in the waterfall model)

System Specification

Requirements Analysis

Architectural Design

Detailed Design

Coding and Debugging

Unit Testing

System Testing

Maintenance

Topics

- Programming techniques and style
- Software design principles
- Basic collaborative programming skills
- Programming tools
- Project-specific subjects

Projects

- 3 projects, each 1 month long
- Each project will be a team project
 - 4 people per team max (max 3 for honors)
- Might require use of specific tools, languages, approaches
- Topics from a wide range of CS fields
 - Lectures will cover additional material

Lab

- Lab times:
 - TA demos/tools instructions
 - Q and A
 - Use as team meeting times
 - Code reviews
 - Right after weekly submissions: Demo your project to the TA (and/or grader) – all teams.

Code Reviews

- Might include code reviews
- Public review/comments on code/design/documentation/etc.
 - During lab or lecture times
- Programs you work on/submit will **not** be considered private, for this class
- You might be asked to present your code

Syllabus Review

- Questions?

About Teamwork

- Working in a team is a major challenge for this course (both for you and for me).
- Look up on the web for info on teamwork.
- Do not slack off. Do not monopolize.
- If you have any issues with your team's dynamic/chemistry, first try to reconcile. Report ASAP if it does not work out. Do not wait until the final project due date.
- Common problems: Not responsive to communication, cannot meet, divisive factions formed.

Common Mistakes

- Writing a lengthy code before compiling and running it.
- Not testing your code.
- Waiting until the last moment.
- Not being proactive: Should have someone start working on parser when this week's submission is DB engine.
- Only one person committing to GitHub.
- Uploaded to eCampus but did not finalize submission.

Plagiarism / Fraud

- We will use MOSS, an automated code plagiarism checker.
 - <https://theory.stanford.edu/~aiken/moss/>
- Will be checked within this semester's submissions and historical files.
- One team found last semester and reported to Aggie Honor System Office and penalized.
- Signatures on attendance sheet must be consistent. Multiple inconsistent signatures were discovered in the previous semesters.

Credits

- Most of the course material for 315 we will use (including syllabus, slides) during this semester has been developed from scratch by Prof. John Keyser.
- Assignments/project details will differ from the past semesters.
- Long Mai and Allen Hurst at Improving Enterprises provided valuable feedback.

To Do

- Download and read this article:
 - Don Knuth's Turing Award Lecture:
 - "Computer Programming as an Art"
 - <http://doi.acm.org/10.1145/361604.361612>
- Read textbook chapters (see weekly schedule for chapters to read each week). There will be two quizzes (online) on the reading material.