CS 121
Software Engineering

Introduction

Why Take This Course?

- Modern software is amazingly large and complex
  - Linux: 12M LoC; Windows: 50M LoC; Google: 2B LoC repo
- How could such large code work at all???
  - Software Engineering (SE)!
    - The study and practice of how to build software
  - Intersects with
    - Programming languages, HCI, management, organizational behavior, …
- Three kinds of SE courses
  - Focus on code ⇐ This course
  - Focus on people ⇐ Other SE courses
  - Focus on “real world” projects ⇐ Capstone project

Important Software Properties

- Correctness
  - The system does what it is supposed to
- Efficiency
  - The system performs its work sufficiently fast
- Maintainability
  - The system can be fixed/changed/improved easily
- Security
  - The system does nothing “bad”
  - Usually means, nothing it is not supposed to
- Reliability
  - The system is robust in expected circumstances
- Other -ilities?

Course Goals

- At the end of this course, you should be able to
  - Program in Java (first topic!)
  - Understand core SE techniques for designing, implementing, testing, debugging, and maintaining code
  - Have the tools and knowledge to build systems 10x–100x larger than in other programming courses
  - Have the base knowledge to learn how to build systems 1,000x–100,000x
    - Note: Every very large software system is its own world, with its own concepts and internal idioms and notations, so each one requires its own study

Topics

- Java programming
- Abstract data types, modularity, information hiding
- Design patterns, including for concurrency
- Software architecture
- Object-oriented refactoring
- Testing
- Concurrency
- Debugging
- (Possibly additional special topics)

Programming Projects

- To be done in Java SE version specified on web page
  - No style guide, but you should try applying ideas from class
- Submitted via Gradescope
- Tip: Make good use of Java libraries!
  - See Java API link on class web page
- Projects due 11:59pm on due date
  - **No projects accepted after that, except:**
  - You have two late tokens for the whole semester, each one giving you 24 hours of additional time to submit a project
- Warning: Don’t expect projects to be perfect
  - We’ll work together to address any issues that arise

Homework

- Might have some written homework assignments
  - Planning not to but want the option just in case
- If we have any, due at **start** of class on due date
- Submitted as pdf on Gradescope

Readings

- Assignments in which you need to
  - Read a paper
  - Answer two questions about the paper
    - A typical answer will be a short paragraph; please don't write a novel
  - Upload pdf with your text to Gradescope
- Due by **start** of class on due date
  - So we can discuss readings in class

- Readings are graded on a scale of
  - 2 - all good!
  - 1 - summary satisfactory but missed key point(s)
  - 0 - not submitted or not satisfactory

Professional Conduct

- Be courteous, professional, and collegial
  - In all interactions with other students and course staff
- This is a higher standard than what is required by law
  - The course instructor is the final arbiter of this standard
- Professional behavior will serve you well as a software engineer
  - Software engineering is a team activity
  - Dysfunctional teams usually create poor software

Grading

- Programming projects/homework (50%)
  - Projects equally weighted
  - If homework assigned, will specify weighting when assignment given out
- Readings (8%)
- Midterm (20%)
- Final (20%)
- Meet your professor (2%)

- Grades posted on Canvas (canvas.tufts.edu)

Textbook

- None

- There is no good book available that covers the right set of topics
  - Use these lecture notes as a reference
  - Take your own notes

Class Videos

- Instructor will try to do screen captures of live sessions
  - Videos posted on Canvas
  - **No guarantee** that videos will work
    - Technical difficulties occasionally might mean no or only partial video for that day
  - You may not share videos outside of class

- **By participating in class, you consent to recording**
  - If you have objections, let instructor know before class
  - Not participating in a recorded class will not affect your grade

Other Administrivia

- Will use Gradescope for all project/homework/ readings submissions
- **Announcements** and discussions on web forum
  - Do not post code or test cases on web forum
    - Unless otherwise permitted
  - Do not give away answers on web forum
- Let me know as soon as possible if you have an excused absence
  - See syllabus for details about excused absences
  - In general, you'll have longer than you need for projects, so you can work around expected issues in your schedule
- Avoid academic dishonesty
  - Do not share code, do not look at others' code, do not post your code on the web

And We're Off!

- Be sure to read and understand the syllabus
  - Don't hesitate to ask if you have any questions
- The course staff is here to help
  - We want everyone in the class to succeed!
- What do you want to get out of the class?
  - Will you have new programming and software design skills?
  - Will you think about software differently at the end of the course?
  - How will you keep learning throughout your career?