COMP 150-SEN
Software Engineering Foundations

Introduction

Spring 2019
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 ⟷ Last semester’s SE course
  ▪ 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 COMP 40
- 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
- Program specification and verification
- Object-oriented refactoring
- Testing
- Debugging
- Program synthesis
- Special topics (TBA)
Programming Projects

• To be done in Java SE 11
  ▪ No style guide, but you should try applying ideas from class

• Submitted via Gradescope

• Projects due 11:59pm on due date
• Projects may be submitted up to 24 hours late for a 10% penalty
• No projects accepted after that

• Warning: All projects will be brand new!
  ▪ So don’t expect them 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
  ▪ If you want to write homework answers by hand, you can use the scanner in CS office
  ▪ But don’t rely on having access at the last minute…
Readings

• Assignments in which you need to
  - Read a paper/blog post/web page
  - Write a 1-2 paragraph summary or answer question(s) specified by assignment
  - 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
First “Reading” Assignment

• Read or watch Tony Hoare’s presentation, *Null References: The Billion Dollar Mistake*
  
  ▪ https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare
  
  ▪ (Okay to read show notes instead of watching)

• Questions to answer
  
  ▪ What was the most interesting thing you learned from the presentation?
  
  ▪ Do you think null pointers really were a billion dollar mistake?

• Due Mon, Jan 28
Grading

• Programming projects/homework (50%)
  ▪ Projects equally weighted
  ▪ If homework assigned, will specify weighting when assignment given out

• Readings (9%)

• Midterm (20%) - tentatively, Wed, Mar 13

• Final (20%) - standard final exam time

• Meet your professor (1%)

• 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

• I will try to do screen captures of lecture
  - Videos posted on Canvas
  - No guarantee that videos will work
    - Technical difficulties occasionally might mean no or only partial video for that day
Other Administrivia

• Will use Gradescope for all project/homework/readings submissions

• **Announcements** and discussions on Piazza
  - Do not post code or test cases on Piazza
  - Do not give away answers on Piazza

• 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