15-122: Principles of Imperative Computation

André Platzer and Anupam Datta

Room change: DH 2210

http://c0.typesafety.net/
Overview

• Goals of this course
• Interactions
  – Lectures, recitations, office hours
• Assessment
  – Quizzes, homework, exams
• A mysterious function!
Computational Thinking

• “Thinking like a computer scientist” is important for lots of people, not just computer scientists!
• A computer science approach to thinking about the correctness of programs
Programming Skills

• Transforming algorithmic ideas to code
• Writing tests
• Imperative programming in C and C0
• Basic Unix survival
Algorithmic Ideas

- Asymptotic complexity
  - time/space/amortized
  - worst case/average case
  - important classes: $O(1)$, $O(\log n)$, $O(n \log n)$, $O(n^k)$, $O(2^n)$

- Big ideas like order and randomness

- Lots of fundamental data structures
  - later used in many advanced computer algorithms
  - (Psst... this is often what tech interviews test on!)
The Big Picture

• Pre- or co-requisites
  – either 15-151 (Math Foundations for CS)
  – or 21-127 (Concepts of Mathematics)

• Counterpart
  – 15-150 (Principles of Functional Programming)

• Pre-requisite for
  – 15-213 (Introduction to Computer Systems)
  – 15-210 (Parallel and Sequential Data Structures and Algorithms)
  – 15-214 (Principles of Software System Construction)
Overview

• Goals of this course
• Interactions
  – Lectures, recitations, office hours
• Assessment
  – Quizzes, homework, exams
• A mysterious function!
Lectures

• Tuesday and Thursday, 9am 📅
• Please be here, please be active
  – Ask and answer questions, pay attention
  – Lecture notes published after lecture
• Laptops for note-taking only
  – No surfing, email, games…
  – If you want to work on your homework, do so elsewhere
  – Too distracting for other students
Recitations

• (Hello, TAs!)
• Wednesday and Friday, starting tomorrow
• Reinforce lecture material and variations
• Problem solving
• How-to programming and tool support
Unix/Tools Tutorial

• Thursday 16th, 7pm-9:00pm, GHC 4401
• Get set up using the C0 tools with Andrew Linux
• Format: drop-in for half an hour
Online communication

- Autolab for homework and grades
- Piazza for announcements, questions, and communication with course staff. Get help, help each other!
- Quiz system for 122: //@Quizzes
- Cluster Linux machines and SSH to shared machines for assignments
Other Resources

• Course home page
  – Schedule, lecture notes, calendar, contact info…
  – Office hours (TBA, starting Friday)
  – Academic development walk-in tutoring

• C0 home page
  – http://c0.typesafety.net/
  – Tutorial, reference, examples, binaries
Overview

• Goals of this course
• Interactions
  – Lectures, recitations, office hours
• Assessment
  – Quizzes, homework, exams
• A mysterious function!
Assessment

• 25% - Midterms (two of them, 12.5% each)
• 25% - Final
• 40% - Weekly Homework
  – Programming due on Monday 10pm, through Autolab
    • 4 late days total, max 1 late day per assignment
    • 50% penalty per day beyond the given late days
  – Written due on Thursday 9am before lecture
    • Late policy: submit by 9am Friday for a ~20% point penalty
      (outside Sammi DiNardo’s office GHC 9118)
• 10% - Weekly Quizzes
  – 10am–10pm Friday
  – 3 lowest scores dropped
Academic integrity

• Quizzes, exams, homework *must be your own*
• *You* must hand in your work
• OK: discussion of course material, practice problems, study sessions, going over *handed-back* homework in groups
• Not OK: copying or discussing or comparing answers, looking at or copying code (even parts)
• Not OK: talking through the assignment as you code with a classmate

• We use MOSS to catch code duplication
• If you make a mistake, come to us, don’t let us come to you
Overview

• Goals of this course
• Interactions
  – Lectures, recitations, office hours
• Assessment
  – Quizzes, homework, exams
• A mysterious function!
A Mysterious Function Approaches!