15-122: Principles of Imperative Computation

André Platzer and Rob Simmons

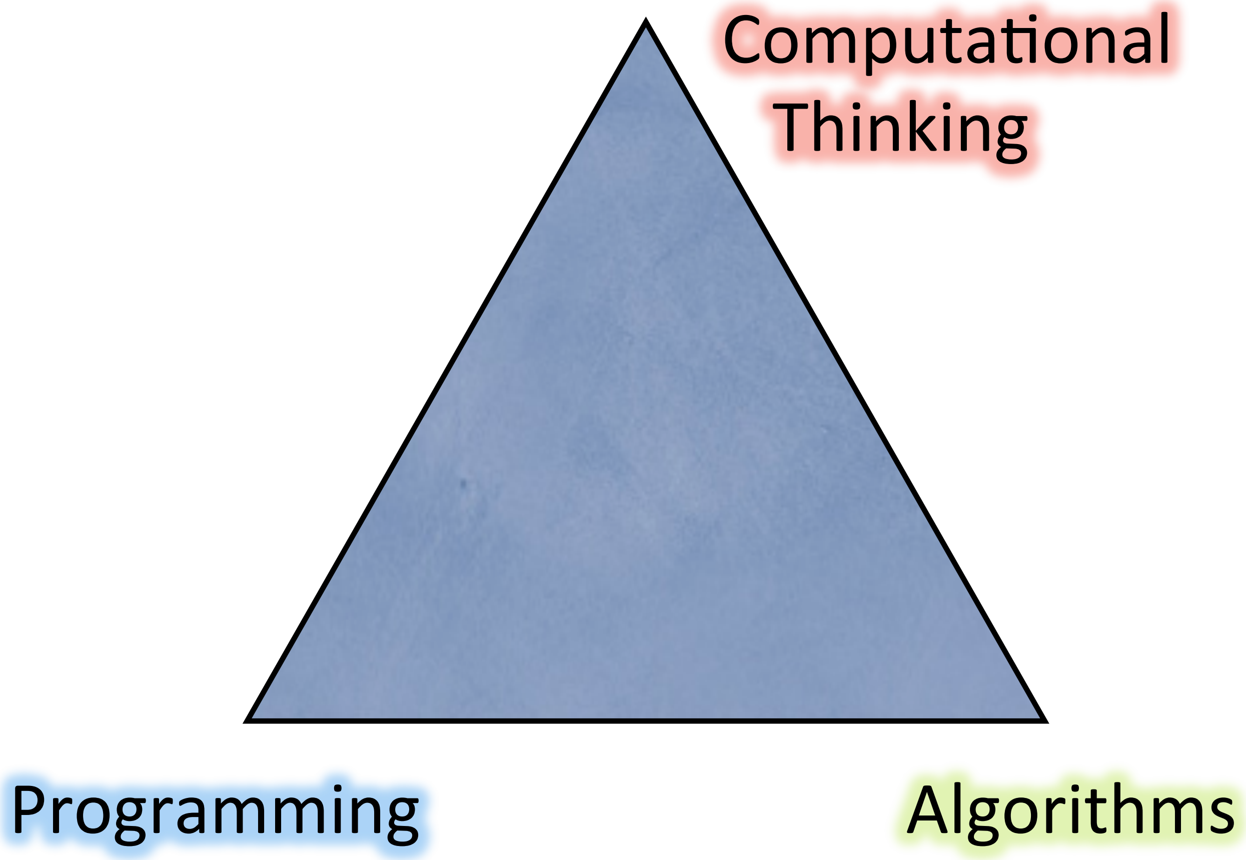
http://symbolaris.com/course/pic13.html

http://c0.typesafety.net/

Overview

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

Goals



Computational Thinking

Programming

Algorithms

Computational Thinking

- Assertions and invariants
- Specification vs. implementation
- Logical vs. operational reasoning
- Abstraction and interfaces
- Reasoning about resource bounds

Programming Skills

- Program design in the small
  - Transforming algorithmic ideas to code
  - Unit testing
  - Specifying, writing, debugging, (re)factoring code
- Some familiarity with Unix tools

Programming Language

- C0: a small safe subset* of C
    - int, bool, char, string, arrays, pointers, structs
- Essential algorithmic and programming ideas
- Close to the machine (imperative)
- Reasoning with contracts
- Transition to C near the end of the course

Algorithmic Ideas

- Asymptotic complexity
  - time/space/amortized
  - worst case/average case
  - important classes: $O(1)$, $O(\log n)$, $O(n \log n)$, $O(n^k)$, $O(2^n)$
- Divide-and-conquer
- Self-adjusting data structures
- Randomness
- Dynamic programming
- Emphasis on imperative programs, ephemeral data structures

Concrete Algorithms

- Basic arithmetic
- Binary search, sorting
- Stacks and queues, priority queues (heaps)
- Binary trees, dictionaries, maps, sets, tries
- Hashing, hash tables
- Graph traversal, minimum spanning tree

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
- Problem solving
- How-to programming and tool support

Unix/Tools Tutorial

- Thursday, 5-7pm, Rashid Auditorium (here)
- Get set up using the C0 tools with Andrew Linux
- Format: drop-in for half an hour
- Makeup sessions in the cluster TBA

Online communication

- Autolab for homework and grades
- Piazza for announcements, questions, and communication with course staff. Get help, help each other!
- Cluster Linux machines and SSH to shared machines for assignments

Other Resources

- Course home page
  - http://symbolaris.com/course/pic13.html
  - Schedule, lecture notes, calendar, contact info…
  - Office hours (TBA, starting Friday)
  - “Lab” hours (TBA, regular time and place in the evening, starting Sunday)
  - 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

- 9% - Quizzes (seven of them, lowest score dropped)
  - Due at midnight(!), check the course schedule
- 20% - Midterms (two of them)
- 25% - Final
- 46% - Homework (nine of them)
  - Combination of written and programming
  - Written due at the BEGINNING of lecture
    - Later on the due date: 5 point penalty (~20%)
    - After due date not accepted w/o advance arrangement
  - Online due at midnight
    - 3 late days total, max 1 late day per assignment
    - 50% penalty per day beyond the given late days

Academic integrity

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

Overview

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

Bug report!