15-122: Principles of Imperative Computation

Iliano Cervesato, Dilsun Kaynar

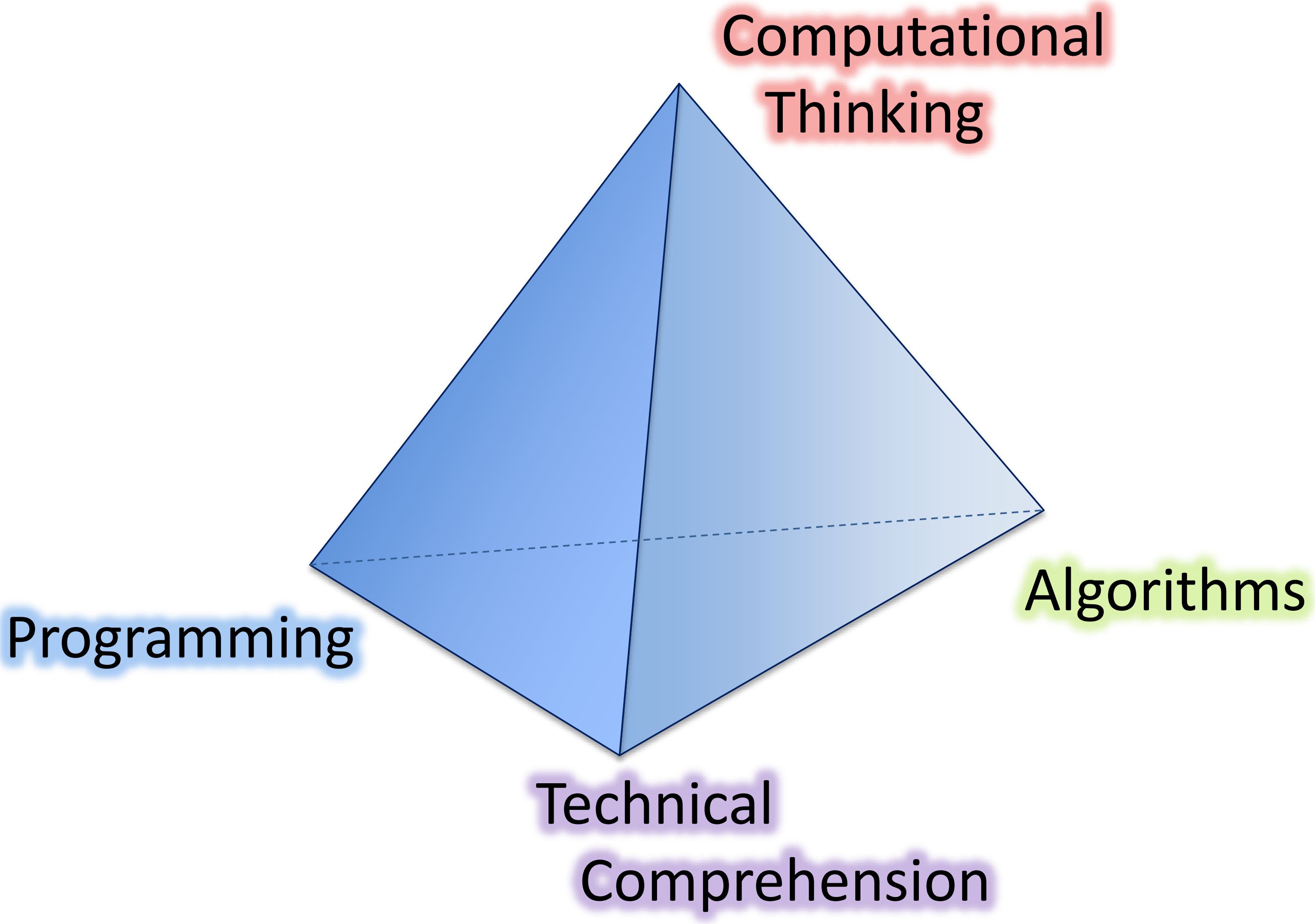
https://cs.cmu.edu/~15122

Overview

- Goals of this course
- Interactions
  - Lectures, labs, recitations, office hours
- Assessment
  - Homework, exams, activities

- The course begins …

Goals



Computational Thinking

Algorithms

Programming

Technical Comprehension

Programming Skills

- Transforming algorithmic ideas into code
  - Code that works the first time around
    - *Deliberate programming*
  - … well, *nearly* the first time around
    - Writing tests
- Imperative programming in C and C0
- Basic Unix skills

Algorithmic Knowhow

- Asymptotic complexity
  - time/space
  - worst case/average case/amortized analysis
  - important classes: *$O(1)$, $O(\log n)$, $O(n \log n)$, $O(n^k)$, …*
- Important ideas like *order* and *randomness*
- Lots of fundamental data structures

(Psst… this is often what tech interviews test on!)

Computational Thinking

- From ***programmer*** to **computer scientist**
  - Systematic approach to solving a problem
  - Finding solutions that are *correct*
  - Finding solutions that are *efficient*

Technical Comprehension

- Learning to read technical specifications is an important skill you will be acquiring in this class
  - – Problem statements will get longer
  - – Dots to be connected will be further apart
  - + You will become more confident
  - + You will try more things on your own

What you will get out of 15-122

- Confidence to write small programs correctly
  - up to a couple thousand lines of code
- Knowledge of lots data structures
  - and algorithms too
- (Some) experience with C
- Systematic approach to solving problems
- Good time management

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
  - Homework, exams, activities
- The course begins …

Lectures

- Tuesdays and Thursdays
- Please be here, please be active
  - ask and answer questions, pay attention
  - lecture notes, slides and online modules for review
- Laptops for note-taking only
  - In the back unless you can’t see the board
    - **Too distracting for other students**
  - No surfing, email, games, homework, …

Labs and Recitations

- Labs Monday (programming exercises)
- Recitations Friday (review & written exercises)

- **Collaborative** problem solving
  - Help others if you are done early!
- How-to programming and tool support
- *Attend the lab/recitation you're registered for*

Getting-started Help

- Laptop setup office hours
    - Set up the C0 tools with Andrew Linux
  - Wednesday, 6 to 8pm in TBA
    - Drop in for half an hour
  - or do it yourself: “C0 at CMU” at https://c0.cs.cmu.edu
- Linux workshops
    - Learn useful Linux commands
  - Thursday 7:30 to 9:30 on Zoom
  - or next Tuesday 4:30 to 6:30 on Zoom

Online Resources

- Course home page http://cs.cmu.edu/~15122
  - Schedule, calendar, contact info…
  - Lecture notes, slides, OLI modules
  - Links to all resources
- C0 home page https://c0.cs.cmu.edu
  - Tutorial, reference, examples, binaries

Online communication

- **Diderot** for announcements, questions, and communication with course staff
  - Get help, help each other!
- **Autolab** and **Gradescope** for homework
- Grades from course home page
- Cluster Linux machines and SSH to shared machines for assignments

Help through the Semester

- **Office hours**
  - Calendar on course web page
- Student Academic Success Center support
  - Supplemental Instruction: TBA
  - Peer Tutoring: TBA

Overview

- Goals of this course
- Interactions
  - Lectures, recitations, office hours
- Assessment
  - Homework, exams, activities
- The course begins …

Assessment

- 50% - Exams (2 midterms and a final)
- 45% - Weekly Homework
  - *Written* due **Monday** by **9pm ET** on **Gradescope**
    - No late days: 50% penalty if handed in by 8am Tuesday
    - ∞ submissions
  - *Programming* due **Thursdays 9pm ET** on **Autolab**
    - Download assignments and code from Autolab or Diderot
    - 3 late days, at most 1 per homework
    - Extensions only for emergencies
- 5% - In-class activities and labs
  - In-class activities in lectures
  - Attend, make a good effort, get credit

Written 1 already Out.
Due this Monday

Academic integrity

- Homework and exams *must be* *<u>your own</u>*
  - ***You are here to learn****, not to get a grade*

- **NOT OK:** discussing hw answers, sharing code
- **OK**: discussing course material, practice problems, blank assignments, study sessions, handed-back homework

If you make a mistake,
come to us, don’t let us come to you

How to do Well in this Course

- Do not stress over grades
- Participate
- Manage your time wisely
  - *Don’t use late days in 1st half of course*
- Start homework early
- Get all the help you need
  - ask for help, tell us when you’re having trouble
- Make time for fun

Overview

- Goals of this course
- Interactions
  - Lectures, recitations, office hours
- Assessment
  - Homework, exams, activities
- The course begins …