Alan Weide

Yale University Dept. of Computer Science

RESOLVE IS USEFUL: LESSONS LEARNED BY A NEW FACULTY MEMBER

Who Am I?

- PhD 2021, The Ohio State University
  - Advisor: Paul Sivilotti
  - Taught CSE 2221: Software Components
- Since 2022, Lecturer of Computer Science, Yale University
  - CPSC 223: Data Structures and Programming Techniques
  - CPSC 419: Full Stack Web Programming

What's in This Talk?

- Reflections of my first year as a full-time faculty member
  - About computer science curricula
  - About students
- A neophyte's (my) thoughts on the role of formal reasoning in introductory computer science courses
  - Do I believe students can learn and apply it?
  - Do I believe the rest of the faculty would accept its introduction into the curriculum?

What Isn't in This Talk?

- Even a modicum of quantitative analysis
- Any actual experience incorporating formal reasoning into a course
- Commentary on the quality of the CS education at OSU, Yale, or anywhere else
- Me talking at you for 30 minutes
  - I have minimal experience with this—let's have a conversation!
  - Please interrupt with arguments, comments, compliments, criticisms, ideas, praises, questions, references, refutations, thoughts, *etc*.

My Reflections

On Computer Science Curricula

- As a student at Ohio State, I had no trouble figuring out what courses I should take and when
- As a faculty member at Yale, I have no idea what courses I should advise my students to take and when
  - Difference in philosophy: Yale is a liberal arts school
    - Students should just take whatever they feel like taking
  - Consequence: Yale courses must accommodate an incredible diversity of student background

On Students

- **Unsurprising**: if it helps them get an 'A', Yale students will do anything you ask of them
  - A story for another time…we asked *a lot* of them this semester
- **Surprising**: Yale students don't claim to "know it already"
  - I am not a web programming expert. My class was incredibly gracious when I gave them bad advice during lectures
- **Surprising**: Yale students are uncomfortable with software engineering, the ideas of reusability and modularity, and are never exposed to function contracts

My Courses

Yale Computer Science Curriculum

- The following are electives at Yale
  - Software engineering
  - Object-oriented programming
  - Programming languages
- The following are required courses at Yale
  - “Introduction to Computer Science” (AKA functional programming)
  - “Introduction to Systems Programming and Computer Organization”

CPSC 223: Data Structures & Pgmming Techniques

- Required for all Computer Science majors at Yale (and CS+X majors)
- Historically taught in C
- Has as prerequisite CPSC 201: Introduction to Computer Science
  - Famously “useless” course
- Prerequisite for CPSC 323: Introduction to Systems Programming & Computer Organization
  - Famously challenging course

CPSC 223: Data Structures & Pgmming Techniques

- Another new lecturer and I are leading a modernization effort
  - C → C & C++
  - Yale compute cluster → cloud platforms (Gradescope)
  - Broad course reorganization (multiple sections)
- Constraints on our reorganization of the course
  - Students must understand C by the time they get to 323
    - Compose nontrivial C programs from scratch
  - We have to teach them the important data structures

CPSC 223: State of the Course

- Programming assignment specs are in English
- Focused on *applications* of data structures
- Submissions are evaluated using a suite of automated tests
  - Unit tests
  - Input/output tests
  - Memory leak detection tests
  - “Performance” tests

CPSC 223: Goals

- Decouple data structures from programming language
- Teach students to reason about software
  - Also: how to build software that they *can* reason about
- Introduce function specs
  - Formal? Informal?
  - C/C++ standard library? RESOLVE/C++? Clean++?
- Evaluation of student software elegance

CPSC 419: Full Stack Web Programming

- New course that I started Fall 2022
  - Directive: create a course that teaches web programming and software engineering in a way that is approachable by non-CS majors
  - This is truly my course—I have essentially complete control
- Technologies
  - Python for server software (Flask web server framework)
  - Relational database (SQLite + SQLAlchemy)
  - JavaScript client software (vanilla JS + jQuery)

CPSC 419: Full Stack Web Programming

- Students build a progressively more complex search application for a curated data set from back end to front end
  - Project 1: Command-line interface (Python + SQLite)
  - Project 2: Native GUI (… + Qt)
  - Project 3: Static web application (… + Flask + HTML)
  - Project 4: Dynamic web application (… + CSS + JavaScript)
- Work with a group to build a web application of their choice over the course of the semester (… + ???)

Discussion

HOW CAN I INCORPORATE FORMAL REASONING INTO MY DATA STRUCTURES COURSE?

CAN I INCORPORATE FORMAL REASONING INTO MY WEB PROGRAMMING COURSE?