

CS 142 Spring 2015

Wrap-Up

Final Exam

- Choose Between:
 - **Monday, May 4th, 1-3:30pm (Regularly scheduled)**
 - Location: **FJ-B (Frazier-Jelke Lecture Hall B)**
 - **Tuesday, May 5th, 8:30-11am (Alternate time)**
 - Location: **Ohlendorf 225**
- Will cover all topics roughly proportionally to the amount of time spent on them in class.
- May involve ideas implemented in programming projects.
- I've provided practice problems as well as solutions for a few, but not all, topics
 - Any questions on practice problems?

Final exam topics

- Exam will be in C++ only (no Python)
- Run-time of algorithms
- C++ intro stuff (not a lot)
- Pass by value, pass by reference
- Strings and vectors (handouts provided on exam)
- Structs
- Pointers and dynamic memory
- Recursion (including binary search algorithm)
 - You will probably have to write a recursive algorithm.
- Classes
- Operator overloading
- Inheritance and polymorphism
- Linked lists

Hint: Keys to the game:
Know what a topic is,
what it's good for, what
it's bad for, how to use
it, and how it relates to
other topics.

Victory Lap

A victory lap is an extra trip
around the track

- By the exhausted victors (us) ☺

Review course goals


- See if we met them

Some big themes and perspectives

- Stuff for five years from now more than for the final



Thank you!

- You all made this a great class
 - Great attitude about learning new CS topics
 - Good class attendance and questions
- Feedback is appreciated on projects, tests, and their respective difficulty (too hard, too easy, just right?)
 
- Did you like learning some concepts in Python first? Or would you have preferred to start with C++ and use it throughout the semester?

What is this class about? (from lecture 1)

- Learning more complex programming concepts.
- Use C++ language.
- Schedule
 - Python to C++ introduction
 - New C++ concepts
 - Object-oriented programming (OOP)

Abstraction

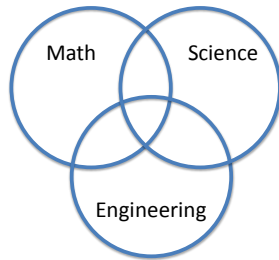
- CS is all about constructing hierarchies of abstractions
 - abstraction = leaving out details that we don't care about to focus on the stuff we do care about.
 - Some abstractions we create for ourselves
 - Some we create for other people using our code.

Abstraction

- Abstractions we've seen in CS142:
 - Structs (an abstraction of a combination of data types)
 - Classes (an abstraction of data types and associated actions)
 - Operator overloading (hiding the details of what it means when you say $v1 = v2 + v3$)

What's next?

- CS 241: Data Structures and Algorithms
- CS 172: Discrete Structures
- CS 231: Computer Organization



Stay In Touch

- Tell me when this class helps you out with something cool (seriously).
- Ask me questions (may not always know the answer, but I can tell you where to find it).
- Don't be a stranger: let me know how the rest of your time at Rhodes (and beyond!) goes... I really do like to know.