CSE 333

Lecture 21 - exam, feedback

Steve Gribble

Department of Computer Science & Engineering

University of Washington

Things to know for the final

C:

- malloc, free, and memory leaks
  - stack vs. heap
- pointers, pointers, and more pointers
  - output parameters
  - function pointers
  - double pointers -- when are they appropriate?

Things to know for the final

C:

- syntax
    - structs, prototypes, typedefs, function pointers
- structure
    - .h vs. .c files
    - header guards
- style
    - consistency: naming, indentation, error checking, ...etc.

Things to know for the final

C++:

- references and how they differ from pointers
- consty-ness and when to use it
- smart pointers
- subclasses
- static vs. virtual dispatch and why it matters
- copy constructors, assignment operators, and how they differ

Things to know for the final

C++:

- templates
- how to use vector, map, list from the STL
- type casting primitives

Things to know for the final

Concepts behind:

- file I/O
- networking (network I/O, DNS names vs. IP addresses, HTTP)
- threading
- multiprocess (fork)

Don’t worry about memorizing all of the APIs, particularly the networking APIs!

Feedback

Lectures

Three styles

- slides talking about concepts
  - e.g., the various styles of concurrency and how they relate
- slides talking about language specifics, mixed with flashes of real code
  - e.g., structs, subclassing, fork, malloc/free
- live coding
  - e.g., using fork to create multiple processes and implications

Exercises

Goals

- get you to write more code, more often, and earlier
- present you with more "blank canvas" problems
- reinforce concepts as we discuss them
- ~1-4 hours per exercise

Mechanisms

- quick turnaround between out, due, and graded
  - coarse-gained grading (0,1) --> (0,1,2,3)
- optional exercises with solutions

Sections

Goal:

- instead of lectures, more lab-like
- guided puzzles and problems
- mandatory attendance

Projects

Goals:

- substantial programming experience
- complex codebase, layered codebase
- do something real, rather than just toy problems

Mechanisms

- mixture of “fill in blank” and “blank canvas”
- provided unit tests, try to pass them
- ~2.5 weeks to do each
- optional teams of two

Exams

Goals

- test some practical skills in constrained timeframe
- forcing function to keep up, learn the material
- provide assessment, grading

Questions for you

- should we have exams at all in this course?
    - take-home exam? in place exam?
- if so, what form?
- if not, how do we get the forcing function?

Discussion board

Goals

- place to seek help for specific problems
- way for us to diagnose how HWs, exercises are going to remedy any major structural issues
- place to discuss class concepts
- quick turnaround from Q to A

Workload

Too little, just right, too much?

- last time I ran the course, workload was too high
- turned the knob down for this offering
    - 3 HWs instead of 4
    - but, mandatory exercises (short)

Anything else?