

CSE 333

Lecture 23 - exam, feedback

Hal Perkins

Department of Computer Science & Engineering

University of Washington



Administrivia

HW4 due tonight

- ▶ Up to 2 late days if you've got them

Exam Friday

- ▶ Topic list posted ~~right after class~~ now
- ▶ Review Q&A in sections tomorrow
- ▶ Do we need additional review late tomorrow (other than planned office hours)?

No office hours planned for Friday

- ▶ Should we think about changing that?

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
- multiprocessing (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 demos with a bit of code tinkering
 - although not too much live coding this time around

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

- tried for quick turnaround between out, due, and graded
 - coarse-grained grading (0,1,2,3)
- optional exercises - no solutions this time, but no requests

Sections

Goal:

- instead of lectures, more lab-like
- guided puzzles and problems
- attendance not officially required, but exercises collected

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 weeks to do each
- optional teams of two on later parts

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

But lightly used this summer

Workload

Too little, just right, too much?

- workload has traditionally been fairly intense
- turned the knob down for this offering
 - ▶ 4 big projects as before, but “add a feature” part of HW4 made optional, not required
 - ▶ slightly fewer exercises

Anything else?