Systems and machine code

- Processors and architecture (last week Monday)
- Assembly (last week Wednesday)
- Function call and return (last week Friday and this week Monday)
- Function pointers (Wednesday)
- Review for final exam (**today**)

Today:

- Broad overview, key concepts
- What kinds of questions to expect

Final exam: Thurs, May 7, 10:30 AM

The final exam will consists in questions like the following:

- A series of problems where you are given a code example and are asked to give examples of various vocabulary terms, as on Test 1. In particular, make sure you know the terms local variable, instance variable, static variable, declaration, initialization, instantiation, and subtype. From the same code example, be able to determine which implementation of a method will be called at runtime.
- A problem where you are given a piece of code and asked to write an invariant for a loop and determine a big-oh category. In particular, the question may require you to think about the interaction between a data structure and an algorithm.
- A problem in which you are asked to add details to a diagram of an address space to describe what happens when a function is called or returns. Make sure in particular that you understand where the ip, rp, sp, and fp registers are pointing in the address space.

- A "plain old" programming problem testing how well you have retained skills from Programming I. This problem will ask you to write or finish writing a method or a class (or both). This problem tests programming skills and knowledge that have been reinforced and reviewed in this course, but does not exercise anything completely new in this course.
- A problem in which you are asked to write a class that implements an interface and reuses another class, similar to a problem on Test 2. This problem might specify that you must use composition, or it might specify that you must use inheritance, or it might give the option, or it might ask you to do both (in two parts). Be prepared to use either composition or inheritance, and make sure you understand the difference.
- A problem in which you are asked to write a C function that uses dynamically allocated memory, similar to a problem on Test 2. Make sure you know how to use `malloc`, `calloc`, and `free`.
- A problem in which you are asked to write a class or series of short classes that uses a design pattern we have studied (State, Strategy, Adapter, or Decorator).

Coming up:

- **Due Fri, May 1.** *Do Project 8, GCD in pseudo-assembly*
  (Late days may not be used on Project 8)