Component testing: stubs, drivers

- Often want to test individual parts of a program in isolation, possibly before the rest of the program has been developed
- May happen in team situations, we’re writing one part, someone else is writing another, we want to test ours
- May happen when we’re writing larger programs, we write some parts before the others, want to test what we’ve written
- Solution: write simple code segments to act as substitutes for the missing parts

Stubs: replacements for things we call

- Suppose a program we're writing needs to call a function, *int foo(float x)*, that hasn't been written yet
- We write our version of foo, that simply prompts the user to enter an integer then returns it
- We run our program, when it calls foo we enter the value foo should have computed, and see if our program does the right thing
- Our version of foo is called a stub, and lets us test our component before the real foo is written

Drivers: replacements for code that calls ours

- Suppose we've written a function that expects two parameters, does some computation and returns the result
- Our function is supposed to be called from some other part of the program, but the program isn't written yet
- We write a program that is just a main routine that prompts the user to enter two values, calls our function, and displays whatever was returned
- This mini-program is called a driver, and allows us to test our function before the real program is written

Driver/stub combinations

- Often we’re working on some intermediate layer of a system, where the part that calls our code isn’t written yet, and some of the things we need to call haven’t been written yet
- To test our component, we need to create a driver and all the necessary stubs
- Many code development/test frameworks help in generating driver/stub templates for component testing, or for automating component testing once the drivers/stubs for a component are ready