Answers to student questions from In-class quiz 4. Spring, 2012

The whatIsX diagnostic quiz question was unclear because f(x) might print something as well.

I see your point! I just edited the wording of that quesiotn so that future students are less likely to be confused. Thanks.

Can we mathematically prove that with i *=2, the runtime is log n, or is it just something to memorize?

It's pretty easy to see. Since we start at 1 and multiply by 2 each tie through the loop, it is easy to see that after k times through the loop, i is 2^k. The loop stops as soon as i >= n. This is the same as 2^k >= n. Taking logs (base 2 of both sides) gives us k >= $\log_2 n$.

Do we need to prove big-oh mathematically?

You need to be *able* to do that, and a couple of questions on WA02 deal with that. We don't have to give a mathematical proof every time involving finding the n_0 and c *every* time we want to use O, theta, or omega. In particular, the ratio of limits property will take us a long way. And we can usually use the informal notions that were developed in 220, since it is easy to use the limit properties to show that they work.

Several questions about the StaticParms example. Two separate things are going on here.   In the Basic class, the *foo* method is overloaded (there are two versions with different parameter types).

This did not happen in the given code but if the line `b.foo(d)` had appeared in main(), it would have printed "In Base.Derived". The (runtime) type of b and the (compile-time) type of d would be used.

But that is not what happens in this program. We instead call whichFoo, whose arguments are both declared to have type base. Consider the call arg1.foo(arg2). The signature of the method to be called is determined at compile time (i.e. statically determined), and arg2 is declared to be of class Base, so it will call the foo(Base) method from whichever class is dynamically chosen. When the call is made, the type of arg1 is base, so the system choose the method from the Base class.

Mark Weiss goes into a little bit more detail in the textbook.