Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-three)

3.1: Prologue

The notion of object-oriented programming has been a powerful influence in the computing world for more than a decade. Java is inherently object-oriented; there's no getting around that. But Java actually derives from a family of languages very strongly influenced by an older tradition: structured programming. Although structured programming first appeared several decades ago, many productive programmers are blissfully unaware of some of its basic tenets. Before we proceed to learn about those mysterious objects, we're going to spend some time talking about some of the notions behind structured programming, because we'll meet them again with different names later on. In the meantime, I'll be able to sneak in a few new bits of Java knowledge that will come in very handy.

Another important feature Java shares with some of its predecessors is the notion of *type safety*, as we discussed last time. This time we're going to see a bit more about how you can use type safety to design better programs.

Some of what follows doesn't really fall under the heading of Structured Programming, or Type Safety, or any lofty phrase other than Good Programming Practice. Java is a language that (to anthropomorphize for a moment) tries enormously hard to get you to follow good practices, and you'll find that your productivity will be greatly enhanced if you try to go along, rather than fight.