Liz Brondos

The Adventures of Super Class Man, Yet Another Superhero

DoME programmer: Help, I'm going crazy trying to keep track of our inventory! Back in the old days, it was easy because all we had to keep track of were books and CDs. Now we have videos, video games, board games, and a bunch of other random stuff! All this code duplication is not good for my program! What am I to do?

Super Class Man: *(flies in from the sky)* Don't worry, I'm here! I'm Super Class Man, here to save you from programming doom!

DoME programmer: Oh, I'm so glad you're here! How can you help me with my program? I have too many classes representing categories of items I need to keep track of!

Super Class Man: You need to learn about inheritance, which allows us to define one class as an extension of another. You need to create a class named Item which will have other classes like CD, Book, Video, etc. as their subclasses.

DoME programmer: What are subclasses?

Super Class Man: Well, a superclass is a class that is extended by another class. This would be your class Item. A subclass is a class that inherits from another class, which would be your CD, Book, and Video classes, for example.

DoME programmer: Oh, all right. So it's kind of like a hierarchy?

Super Class Man: Exactly! In fact, classes are linked through inheritance relationships from an inheritance hierarchy. The diagram looks kind of like a family tree.

DoME programmer: Oh, I see. How do constructors work with superclasses and subclasses then?

Super Class Man: Well, the constructor of a subclass must always invoke the constructor of its superclass as its first statement. Otherwise, Java will try to insert a call automatically, but this only works if the superclass has a constructor without parameters.

DoME programmer: Wow, superclasses and subclasses sound like the answer to all my problems! I can reuse previously written classes in a new context!

Super Class Man: Yes, you can. But before you can start using object inheritance, you should know about subtypes. Types form a type hierarchy, and the type defined by a subclass definition is a subtype of the type of its superclass. Variables can hold objects of their declared type or of any subtype of their declared type. For example, if we were to create a superclass Superhero with subclasses Superman, Spiderman, and Super Class Man, we could create a variable of type Superhero which could hold an object of type Super Class Man, and this would be perfectly legal. Also, subtype objects can be used wherever objects of a supertype are expected. This is called substitution.

DoME programmer: I see.

Super Class Man: Oh, and one more thing. All classes with no explicit superclass have Object as their superclass. So you could say that Object is the superclass of every class under the sun except the Object class itself.

DoME programmer: Oh, so you mean it's like Plato and his universal theory of Forms?

Super Class Man: Uhhh…I don't understand philosophy very well, but whatever helps you to understand the concepts better works.

DoME programmer: Well, you need to pick up a book every once in awhile! Here, let me look for a book on philosophy for you. Oh wait, my program is not compiling! I made some changes and forgot to change all the instances of code duplication.
Super Class Man: Well, go change your program using object inheritance like I taught you, before you go out of business!
DoME programmer: All right, I will.
Super Class Man: Until then, farewell! *(Flies away.)*
DoME programmer: No…wait! I wanted to teach you about Plato! Oh well.