Java Object-Oriented Concepts Unit

Lesson 4: Using Objects

SOFTWARE GUILD

The Learning House

427 S 4th Street #300

Louisville KY 40202

Lesson 4: Using Objects

Overview

In this lesson, we'll review the concepts of good class design, including cohesion, composition, and method overloading.

Cohesion

As discussed in earlier lessons, well-designed classes are cohesive. This means that they do one job and they do it well. When designing classes, make sure that you don't think too small or too big. If your classes are too narrow, you end up with many little classes that have to be used together to accomplish anything of significance. On the other hand, if your classes are too broad, you end up with one or two classes that cover several areas of responsibility — these classes tend to be unfocused, sprawling, and usually don't do anything particularly well.

Composition

One of the five attributes of an object-oriented language (according to Alan Kay) is that objects can be made up of other objects. This is known as composition. Composition helps with cohesion and encapsulation by delegating responsibility that falls outside of your class's area of expertise to another object. This is a great way to add capabilities to your class without having to write a lot of code. The object or objects that your classes use behind the scenes are part of your **private implementation** — clients of your class have no need to know how your class gets its work done.

Method Overloading

Method overloading is a language feature that allows us to have several methods that have the same name. In order for this to work, each method must have a different method signature. If each of the methods has the same name, this means that the parameter list must be different for each of these methods.

We have seen several examples of this, including the System.out.println(...) method. There is a version of println(...) that takes a String parameter, another that takes an int parameter, another that takes a float parameter, and many more. The advantage of overloading this method name is that clients only have to remember the "println(...)" method name and can simply pass it whatever type they want to print to the Console.