What is Object Oriented Programming?

- Class = Attributes + Methods
- Objects are instances of a class

Notes:

In classic, procedural programming you try to make the real world problem you're attempting to solve fit a few, pre-determined data types: integers, floats, Strings, and arrays perhaps. In object oriented programming you create a model for a real world system. Classes are programmer-defined types that model the parts of the system.

A *class* is a programmer defined type that serves as a blueprint for instances of the class. You can still have ints, floats, Strings, and arrays; but you can also have cars, motorcycles, people, buildings, clouds, dogs, angles, students, courses, bank accounts, and any other type that's important to your problem.

Classes specify the data and behavior possessed both by themselves and by the objects built from them. A class has two parts: the fields (or attributes) and the methods. Fields describe what the class is. Methods describe what the class does.

Using the blueprint provided by a class, you can create any number of objects, each of which is called an *instance* of the class. Different objects of the same class have the same fields and methods, but the values of the fields will in general differ. For example, all humans have eye color but the color of each human's eyes can be different from others.

On the other hand, objects have the same methods as all other objects in the class except in so far as the methods depend on the value of the fields and arguments to the method.

This dichotomy is reflected in the runtime form of objects. Every object has a separate block of memory to store its fields, but the bytes in the actual methods are shared between all objects in a class.

Another common analogy is that a class is to an object as a cookie cutter is to a cookie. One cookie cutter can make many cookies. There may be only one class, but there can be many objects in that class. Each object is an instance of one class.

Previous | Next | Top

Last Modified September 1, 1997