

KEY KEY KEY KEY KEY KEY KEY KEY KEY KEY KEY KEY KEY KEY

Java Objects Fill-in-the-blank #1: Donuts and Cars

Part A: Instantiation

In Java, the keyword new instructs the jvm to create an instance of the specified class in memory. Each new object we construct is made with its own set of the member variables and method specified on the blueprint class. In our Donut example, each instance of our Donut class had two Member variables: String name and int percentRemaining.

Part B: Blueprint and client classes

Our java programs now involve two or more classes of our own design: one we call the blueprint class because it acts as an instruction sheet for constructing objects of that class. Blueprint classes will NOT contain the program's main method, and therefore cannot stand alone as a working Java program.

It needs a partner class! The second kind of class in our object-oriented programs acts as the client of our blueprint class. Just like the client of a business uses that business's services to solve a problem or carry out a task, our client class uses the member variables and methods of our blueprint class to carry out a programming endeavor, such as simulating a factory or a repair shop.

Unlike our blueprint classes, our client class contains the program's main method. In this method, we construct objects using a reference to our blueprint class and the new keyword. Once we create the object, we store its location in a special variable called a reference variable, also called a pointer variable.

We can then use the magical (small but mighty) dot operator to access member variables and methods located on our newly created objects

Part C: Static vs. Instance

When we create a blueprint class that we intend on instantiating, we do not use the modifier static when declaring member variables and methods. In other words, the static modifier could be interpreted to mean "we won't be creating an object out of this class".