

CLASSES AND OBJECTS IN JAVA

Java is a programming language that follows the Object-Oriented Programming (OOP) principles. OOP model focusses on realworld objects to model applications. A real world object will have some properties and some behaviour. For example, human will have properties like height, weight etc. and behaviours like walk, run talk, etc.

Classes vs. Objects

Classes (and objects) in Java are used to model these real world objects. A class will have a set of properties and a set of behaviours. Properties are represented as variables that hold some values and behaviours are represented as methods (similar to functions in other programming languages).

A class is a blueprint and object is a real representation of that blueprint. For example, Human class may define the properties of a human like height, weight etc. Each individual like you and me are instances (or objects) of that class with specific and different values for these properties.

Clearing a class

We can create a Class as:

```
class Hello {  
  
}
```

- Here, class is a keyword, and is followed by a name for your class, and an opening and closing brace to set the boundaries for this class.
- We can add the optional access modifiers (e.g. **public** class Hello {}) to the class, which specifies the scope of the class.
- We need to save this class in a .java file, which is the file extension for java source code file.

Rules for declaring classes in a Java file

There are some simple rules for declaring classes in a java file:

1. There can be only one public class per source code file.
2. If there is a public class in a file, the name of the file must match the name of the public class and it is case sensitive.

3. A file can have more than one non-public class.
4. Files with no public classes can have a name that does not match any of the classes in the file.

Properties and behavior of a class

There can be two types of members at class level (outside of all methods, but inside of class boundaries {}): instance members and static members.

- Each object created from a class will have its own copies of instance variables.
- Static members belong to the class and there will be only one copy of static members shared by all objects.
 - Static members have static mentioned along with their declaration.
- Class members can be variables and methods; they are explained in future tutorials.

Example

```
class MyClass{  
  
    int myInsVar1; //variable  
  
    int myInsVar2; //variable  
  
    static int myStaticVar1;  
  
    static int myStaticVar2;  
  
    public void myMethod1() {  
  
        //some data  
  
    }  
  
    public void myMethod2() {  
  
        //some data  
  
    }  
  
}
```

Creating objects

Objects of a class are created using the new keyword followed by a constructor name.

```
MyClass myClassVar = new MyClass();
```

- A constructor is a special method used for initializing an object, with the same name of the class, and without a return type (unlike a method).
- If you do not create a constructor, Java will create a default constructor that accepts no parameters and does nothing (empty body).
- With the use of new keyword followed by a constructor (new MyClass()), Java allocates memory for the newly created object in the heap memory and return a reference to it which is stored in a reference variable matching the type.

The Object class

All classes in Java are actually direct or indirect sub classes of the java.lang.Object class directly or indirectly. Hence methods of the Object class are inherited by all classes in Java. You will read more about the Object class.

Source : <http://javajee.com/classes-and-objects-in-java>