**OBJECT AND CLASS IN JAVA**

1. [Object in Java](http://www.javatpoint.com/object-and-class-in-java#object)
2. [Class in Java](http://www.javatpoint.com/object-and-class-in-java#class)
3. [Instance Variable in Java](http://www.javatpoint.com/object-and-class-in-java#objectinstancevariable)
4. [Method in Java](http://www.javatpoint.com/object-and-class-in-java#objectmethod)

In this page, we will learn about java objects and classes. In object-oriented programming technique, we design a program using objects and classes.

Object is the physical as well as logical entity whereas class is the logical entity only.

**OBJECT IN JAVA**

An entity that has state and behavior is known as an object e.g. chair, bike, marker, pen, table, car etc. It can be physical or logical (tangible and intangible). The example of intangible object is banking system.

An object has three characteristics:

* **state:** represents data (value) of an object.
* **behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
* **identity:** Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But, it is used internally by the JVM to identify each object uniquely.

For Example: Pen is an object. Its name is Reynolds, color is white etc. known as its state. It is used to write, so writing is its behavior.

**Object is an instance of a class.** Class is a template or blueprint from which objects are created. So, object is the instance (result) of a class.

**Object Definitions:**

* Object is *a real world entity*.
* Object is *a run time entity*.
* Object is *an entity which has state and behavior*.
* Object is *an instance of a class*.

**CLASS IN JAVA**

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

* **fields**
* **methods**
* **constructors**
* **blocks**
* **nested class and interface**

Syntax to declare a class:

1. **class** <class\_name>{
2. field;
3. method;
4. }

**Instance variable in Java**

A variable which is created inside the class but outside the method, is known as instance variable. Instance variable doesn't get memory at compile time. It gets memory at run time when object (instance) is created. That is why, it is known as instance variable.

**METHOD IN JAVA**

In java, a method is like function i.e. used to expose behavior of an object.

Advantage of Method

* Code Reusability
* Code Optimization

*new* keyword in Java

The *new* keyword is used to allocate memory at run time. All objects get memory in Heap memory area.

Object and Class Example: main within class

In this example, we have created a Student class that have two data members id and name. We are creating the object of the Student class by new keyword and printing the objects value.

Here, we are creating main() method inside the class.

*File: Student.java*

1. **class** Student{
2. **int** id;//field or data member or instance variable
3. String name;
5. **public** **static** **void** main(String args[]){
6. Student s1=**new** Student();//creating an object of Student
7. System.out.println(s1.id);//accessing member through reference variable
8. System.out.println(s1.name);
9. }
10. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=Student)

Output:

0

null

Object and Class Example: main outside class

In real time development, we create classes and use it from another class. It is a better approach than previous one. Let's see a simple example, where we are having main() method in another class.

We can have multiple classes in different java files or single java file. If you define multiple classes in a single java source file, it is a good idea to save the file name with the class name which has main() method.

*File: TestStudent1.java*

1. **class** Student{
2. **int** id;
3. String name;
4. }
5. **class** TestStudent1{
6. **public** **static** **void** main(String args[]){
7. Student s1=**new** Student();
8. System.out.println(s1.id);
9. System.out.println(s1.name);
10. }
11. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestStudent1)

Output:

0

null