

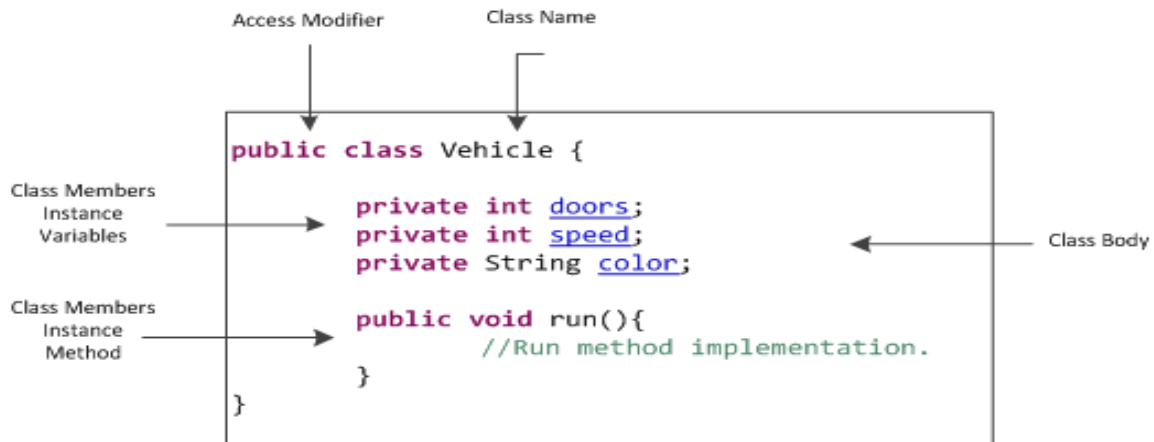
JAVA CLASS, METHODS, INSTANCE VARIABLES

Java Declaration and Access Modifiers

All computer programs consist of two elements: code and data. Furthermore, a program can be conceptually organized around its code or around its data. The first way is called process oriented model. Procedural languages such as C employ this model to considerable success. To manage increasing complexity the second approach called object oriented programming was conceived. An object oriented program can be characterized as data controlling access to code. Java is object oriented programming language. Java classes consist of variables and methods (also known as instance members). Java variables are two types either primitive types or reference types. First let us discuss how to declare class, variables and methods then we will discuss about access modifiers.

Declaration of Class:

A class is declared by use of the class keyword. Class body is enclosed between curly braces {and}. The data, or variables, defined within a class are called instance variables. The code is contained within methods. Collectively, the methods and variables defined within a class are called members of the class.



Declaration of Instance Variables:

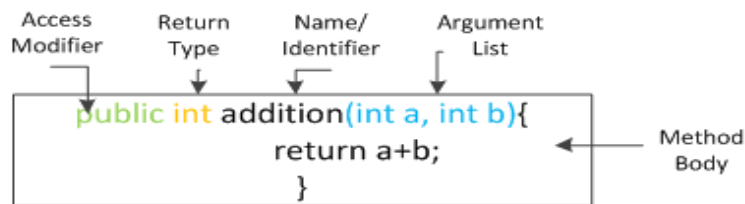
Variables defined within a class are called instance variables because each instance of the class (that is, each object of the class) contains its own copy of these variables. Thus, the data for one object is separate and unique from the data for another. Instance variable can be declared public or private or default (no modifier). When we do not want our variable's value to be changed outside our class we should declare them private. public variables can be access and changed from outside of the class. We will have more information in OOP concept tutorial. Syntax is shown below.



Declaration of Methods:

A method is a program module that contains a series of statements that carry out a task.

To execute a method, you invoke or call it from another method; the calling method makes a method call, which invokes the called method. Any class can contain an unlimited number of methods, and each method can be called an unlimited number of times. The syntax to declare method is given below.



Access modifiers:

Each object has members (members can be variable and methods) which can be declared to have specific access. Java has 4 access level and 3 access modifiers. Access levels are listed below in least to most restrictive order.

public: Members (variables, methods and constructors) declared public (least restrictive) within a public class are visible to any class in the Java program, whether these classes are in the same package or in another package. Below screen shot shows eclipse view of public class with public members.

```
1 package sct;
2
3 public class Vehicle {
4
5     public int doors;
6     public int speed;
7     public String color;
8
9     public void run(){
10         //Run method implementation.
11     }
12
13 }
14
```

protected: The protected fields or methods cannot be used for classes and Interfaces. Fields, methods and constructors declared protected in a super class can be accessed only by subclasses in other packages. Classes in the same package can also access protected fields, methods and constructors as well, even if they are not a subclass of the protected member's class.

```
1 package sct;
2
3 protected class Vehicle {
4
5     protected int doors;
6     protected int speed;
7     protected String color;
8
9     protected void run(){
10         //Run method implementation.
11     }
12
13 }
14
```

Default (no value):The default access level is declared by not writing any access modifier at all. Any class, field, method or constructor that has no declared access modifier is accessible only by classes in the same package.

```
Vehicle.java x
1 package sct;
2
3 class Vehicle {
4
5     int doors;
6     int speed;
7     String color;
8
9     void run(){
10         //Run method implementation.
11     }
12
13 }
```

private: The private (most restrictive) modifiers can be used for members but cannot be used for classes and Interfaces. Fields, methods or constructors declared private are strictly controlled, which means they cannot be accessed by anywhere outside the enclosing class.

```
Vehicle.java x
1 package sct;
2
3 public class Vehicle {
4
5     private int doors;
6     private int speed;
7     private String color;
8
9     public void run(){
10         //Run method implementation.
11     }
12
13 }
```

Java has modifiers other than access modifiers listed below:

static: static can be used for members of class. The static members of class can be access without creating object of class. Lets take an example of Vehicle class which has run () as static method and stop () as non-static method. In Maruti class we can see how to access static method run () and non-static method stop ().

```
Vehicle.java X
1 package sct;
2
3 public class Vehicle {
4
5     private int doors;
6     private int speed;
7     private String color;
8
9     public static void run(){
10         //Static Run method implementation.
11     }
12
13     public void stop (){
14         //Implementation of Stop method
15     }
16 }
17
18 class Maruti {
19     public void TestVehicleClass(){
20         //To Access run() method we dont need object of Vehicle class
21         Vehicle.run();
22         // To Access stop() method we need object of Vehicle class, else compilation fails.
23         new Vehicle().stop();
24     }
25 }
```

final: This modifier applicable to class, method and variables. This modifier tells compiler not to change value of variable once assigned. If applied to class, it cannot be sub-classed. If applied to method, the method cannot be over-ridden in sub-class. In below sample we can see compiler errors while trying to change value of filed age because it is defined as final while we can change value of name field.

```
Vehicle.java *Person.java X
1 package sct;
2
3 public class Person {
4     private String name="Amit";
5     private final int age= 30 ;
6
7     public void finalDemo(){
8         name = "Amit Himani";
9         age=35;
10    }
11
12 }
13
```

The final field Person.age cannot be assigned
1 quick fix available:
Remove 'final' modifier of 'age'
Press 'F2' for focus

abstract: There are situations in which you will want to define a super class that declares the structure of a given abstraction without providing a complete implementation of every

method. This modifier is applicable to class and methods only. We will discuss abstract class in detail in separate Tutorial.

Below Table summarizes the access modifiers

Modifier	class	constructor	method	Data/variables
public	Yes	Yes	Yes	Yes
protected		Yes	Yes	Yes
default	Yes	Yes	Yes	Yes
private		Yes	Yes	Yes
static			Yes	
final	Yes		Yes	

Let's take first column example to interpret. A "class" can have public, default, final and abstract access modifiers.

Summary

- Access modifiers help to implement encapsulation principle of object orientation programming.
- Java has 4 access modifiers public, protected, default, private.
- Java has other modifiers like static, final and abstract.

Source: <http://www.w3resource.com/java-tutorial/java-class-methods-instance-variables.php>