

A class is a template for creating objects which define its state and behaviour. A class contains field and method to define the state and behaviour of its object.

Syntax for Declaring Class:

```
<Access Modifier> class <Class_Name> extends <Super_Class_Name> implements  
<Interface_Name>
```

ACCESS MODIFIER:

Defines who in the Java world can access this class and members of the class.

CLASS_NAME: Unique name for the class in a specific package.

SUPER_CLASS_NAME: Name of the class which given class extends. (extends keyword is used for this purpose)

INTERFACE_NAME: Name of an Interface which above class implements. (implements keyword is used for this purpose)

The internal structure of Class

```
<Access_Modifier> class <Class_Name> extends <Super_Class_Name>  
implements <Interface_Name>{  
    <static initilizar block>  
    <ananymous_block>  
    <constructor_declarations>  
    <field_declarations (Static or Non-Static)>  
    <method_declarations (Static or Non-Static)>  
    <Inner_class_declarations>  
    <nested_interface_declarations>  
    variables_inside_class(Static or Non Static)  
}
```

Example of JAVA CLASS:-

```
package com.jbt;  
  
import java.lang.*;
```

`/* As this file contains public class. Then the name of this file should be TestClass.java */`

```
public class TestClass {  
    public int i; // This is Non Static variable  
    static {  
        System.out.println("This is static block of our class ");  
    }  
  
    {  
        System.out.println("This is anonymous block");  
    }  
  
    TestClass() {  
        System.out.println("This is constructor of the class ");  
    }  
    void method() {  
        System.out.println("This is a normal method");  
    }  
} //close of class named TestClass
```

`//Another class as per our need`

```
class AnotherClass {  
}
```

Classes are written in a Java source file. A source file can contain more than one Java Class. There are some rules associated to the Java source file as listed below.

Rules Applicable to Source file

- There can be only one public class per source code file but it can have multiple non-public classes.
- In case there is any public class present in the source code file, the name of the file should be the name of the class.
- The sequence of statements in a source code file should be package >> import >> Class declaration.
- No Sequence rule is applied for Comments.
- Comments can be there in any part of the source code file at any location.
- Files with no public class can have any name for the class.
- Import and package statements should be applied to all the classes in the same source code file.

Object

There is often confusion about the difference between Class and Object. A class is the creation of a prototype and Object is the actual implementation of that prototype. Technically, Class is a template that describes what state and behaviour an instance of this class can have. Object implements the state and behaviour in the form of variables and methods and requires some memory allocated.

How to Create an Object of a Class

To create an Object of a Class <new> Keyword can be used.

Syntax:

```
<Class_Name> ClassObjectReference = new <Class_Name>();
```

Here the constructor of the Class(Class_Name) will be executed and an Object will be created(ClassObjectReference will hold the reference of the created Object in memory).

Example

```
package com.jbt;

public class HelloWorld {

    int i; // Class Variable(State)

    void method() {

        System.out.println("Inside Method");

    } // Method (Behavior)

}
```

Object Create

```
HelloWorld obj = new HelloWorld();
```

How to Access the Member of a Class

(ClassObjectReference.member). You will call the Method of an Object by naming the Object followed by a period (dot), followed by the name of the Method and its argument list.

```
objectName.methodName(arg1, arg2, arg3);
```

Memorable Points

- A class can have only public and default access.
- The public class needs to be in the same name java file.
- A single java file can contain more than one non-public class but can have only one public class.
- A public class can be seen by all classes from all packages
- A class with default access can be seen only by classes within the same package.
- A Java file with no public class has no naming restriction.
- The Class can also have final, abstract, and strictfp non-access modifiers.
- An Abstract Class cannot be instantiated.
- A Final Class cannot be subclassed.
- A Class cannot be both final and abstract.
- Class visibility can be seen in 3 parameter
 1. If a class can extend another Class?
 2. If a class can create an instance of another Class?

3. If a Class can access the methods and variables of another Class?

Please ask any queries in whatsapp group. In the next class :→ Constructors