

Sunday Commin topics :

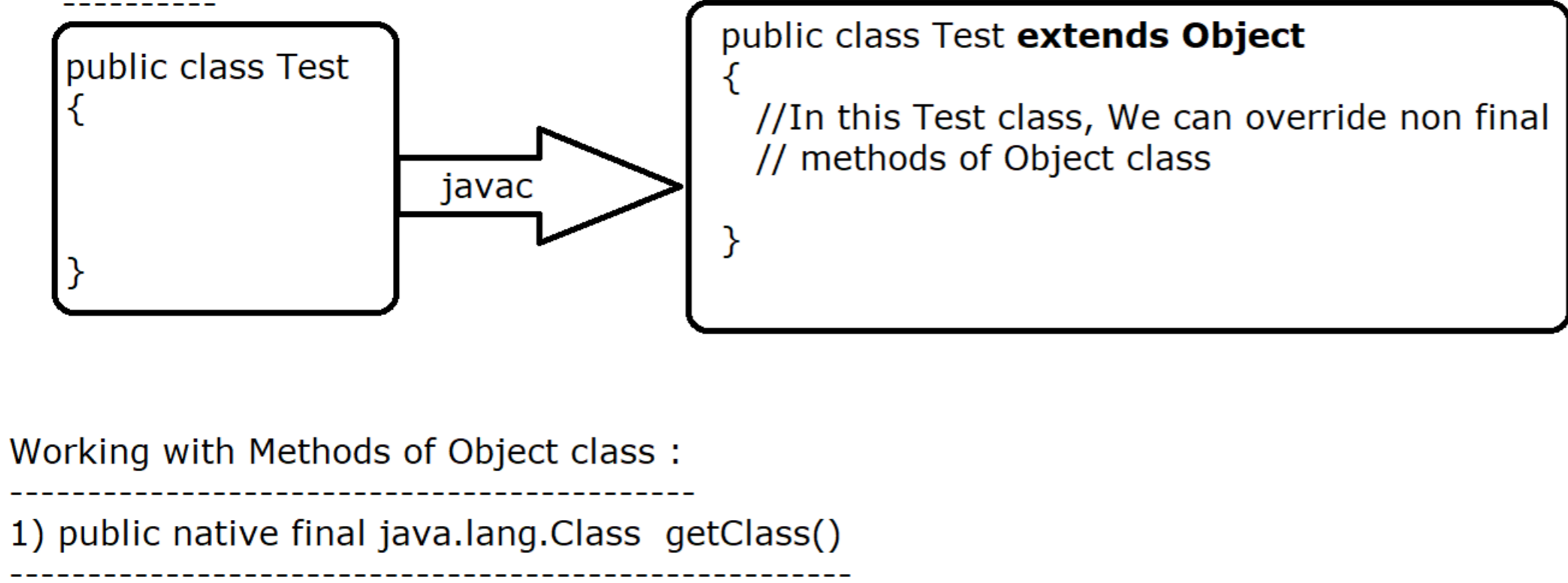
- 1) Object class and It's Methods
- 2) Nested class OR Inner class
- 3) Enum in Java
- 4) Input and Output, File Handling

Object class and It's Methods :

* Object is a predefined class available in java.lang package and It is the super class of all the classes we have in java.

* Object class contains 11 non static methods and 1 no argument constructor.

* Since Object is the super class for all the classes hence we can override the Object class methods (non final) in the sub class.



Working with Methods of Object class :

1) public native final java.lang.Class getClass()

* It is a predefined **non static final** method of Object class.

* It is used to provide runtime information of a class that means it will provide complete information of a class which is as follows :

class keyword + Fully Qualified Name (Package Name + class Name)

```
//Programs
package com.ravi.object_methods;

class Test
{
}

public class GetClassDemo1
{
    public static void main(String[] args)
    {
        Test t1 = new Test();
        System.out.println(t1.getClass());

        String s1 = "Java";
        System.out.println(s1.getClass());

        Integer i1 = 90;
        System.out.println(i1.getClass());
    }
}
```

Note : class keyword + Fully Qualified Name.

java.lang.Class class provides a non static method called getName() which provides Fully Qualified name

```
public String getName();

package com.ravi.object_methods;

public class GetClassDemo2
{
    public static void main(String[] args)
    {
        Double d1 = 23D;
        String fqcn = d1.getClass().getName();
        System.out.println("Fully Qualified Name is :"+fqcn);
    }
}
```

2) public native int hashCode() :

* It is a predefined non static method of Object class.

* The main purpose of hashCode() method to find out the **bucket location in the Hashtable data structure**.

* The return type of this method is int so, It returns the numeric value to store the data in the bucket.

* WE SHOULD NEVER COMPARE TWO OBJECTS BY USING hashCode(), We can compare by using equals(Object obj) method.

* **If two objects are equal by using equals(Object obj) method then hashCode of both the Object must be same.**

//WAP to show that two different objects can have same hash code :

```
package com.ravi.object_methods;

public class HashCodeDemo1 {

    public static void main(String[] args)
    {
        String s1 = "A";
        Integer i1 = 65;

        System.out.println(s1.hashCode()); //65
        System.out.println(i1.hashCode()); //65
    }
}

package com.ravi.object_methods;

public class HashCodeDemo2
{
    public static void main(String[] args)
    {
        String str1 = "india";
        String str2 = new String("india");

        System.out.println(str1== str2); //false
        System.out.println(str1.equals(str2)); //true

        System.out.println(str1.hashCode());
        System.out.println(str2.hashCode());
    }
}
```

* **There is a contract between hashCode() and equals(Object obj) method, The contract says we should always override both the methods in any class.**

public String toString() :

* It is a predefined non static methods of Object class.

* It is used to provide **"String representation of the Object"**.

* Once the object is represented in the String format then It is very easy and concise to read the information.

* In order to display non static field we should override this method in the sub class.

```
Logic Written by Java software people in toString() Method :

public String toString()
{
    return getClass().getName()+"@"+Integer.toHexString(hashCode());
}

Note : toString() method internally uses getClass() and hashCode() method of Object class

package com.ravi.object_methods;

class Foo
{
}

public class ToStringDemo {

    public static void main(String[] args)
    {
        Foo f1 = new Foo();
        System.out.println(f1.toString()); //com.ravi.object_methods.Foo@5acf9800

        System.out.println("=====");

        String name = f1.getClass().getName();
        int hash = f1.hashCode();

        System.out.println(name+"@"+Integer.toHexString(hash)); //com.ravi.object_methods.Foo@5acf9800
    }
}
```

```
package com.ravi.object_methods;

class Student
{
    private int id;
    private String name;

    @Override
    public String toString()
    {
        return "Student [id=" + id + ", name=" + name + "]";
    }
}

public class ToStringDemo1
{
    public static void main(String[] args)
    {
        Student s1 = new Student();
        System.out.println(s1);

        Student s2 = new Student();
        System.out.println(s2);
    }
}
```

public boolean equals(Object obj) :