Object class in java

Object Class –

It is the parent class of all the classes in java. It is called as topmost class of java which is present in java.lang package. There are different methods of object class are as follows.

1. Public final Class getClass()

This class is used to get the metadata of class.

```
package com.wipro.jpmorgan;

public class Example {

    public static void main(String[] args) {

    Example example = new Example();
        System.out.println(example.getClass().getName());
        System.out.println(example.getClass().getSimpleName());

    }
}
```

2. Public int hashCode()

For every object unique number is generated by JVM called as hashcode. It does not represent object of address then what is the use of hashcode.

HashCode in Java is a function that returns the hashcode value of an object on calling. It returns an integer or a 4 bytes value which is generated by the hashing algorithm.

Note- if two objects are equal, their hashcode will be same.

If two object hashcode are same, you cannot guaranty those objects are equal.

```
package com.wipro.jpmorgan;

public class Example {

    public static void main(String[] args) {

        Example example1 = new Example();
        Example example2 = new Example();

        System.out.println(example1.hashCode());
        System.out.println(example2.hashCode());
    }
}
```

3. Public Boolean equals (Object obj)-

It compares the given object to this object. There are two equals method, this equals method is used to check the address of string not contents.

```java
package com.velocity;

public class Employee {

    int empId;
    String empName;

    public static void main(String[] args)
    {
        Employee emp1 = new Employee();
        emp1.empId = 1;
        emp1.empName = "Ashok";

        Employee emp2 = new Employee();
        emp2.empId = 2;
        emp2.empName = "Sachin";

        System.out.println(emp1.equals(emp2));
    }
}
```

4. protected Object clone() throws CloneNotSupportedException

It creates and returns the exact copy (clone) of this object.

```java
package com.wipro.jpmorgan;

public class Example implements Cloneable {

    int x;

    public static void main(String[] args) throws CloneNotSupportedException {

        Example example1 = new Example();
        example1.x = 50;

        System.out.println("First Object data is>>" + example1.x);

        Object example2 = example1.clone();

        System.out.println("Second Object data is>>" + example2);
    }
}
```

5. public String toString() –

It returns the string representation of this object.

```java
package com.wipro.jpmorgan;

public class Example {

    int x;

    @Override
    public String toString() {
        return "Example [x=" + x + "]";
    }

    public static void main(String[] args) throws CloneNotSupportedException {

        Example example1 = new Example();
        example1.x = 50;

        System.out.println("First Object data is>>" + example1);

    }
}
```

6. public final void notify()-

It wakes up single thread, waiting on this object's monitor.

7. public final void notifyAll()-

It wakes up all the threads, waiting on this object's monitor.

8. public final void wait(long timeout)throws InterruptedException()-

It causes the current thread to wait for the specified milliseconds, until another thread notifies (invokes notify() or notifyAll() method).

9.public final void wait(long timeout,int nanos)throws InterruptedException-

It causes the current thread to wait for the specified milliseconds and nanoseconds, until another thread notifies (invokes notify() or notifyAll() method).

10. public final void wait()throws InterruptedException

It causes the current thread to wait, until another thread notifies (invokes notify() or notifyAll() method).

11. protected void finalize()throws Throwable

It is invoked by the garbage collector before object is being garbage collected.

Wrapper class in java

It provides the mechanism to convert primitive's data type into object and object into primitive data type called as wrapper class.

Process of converting primitive's data type into object called as "Autoboxing." And process of converting object into primitive's data type called as "Unboxing."

There are 8 classes of java.lang.package are known as wrapper classes in java.

Integer

Short

Byte

Long

Double

Character

Boolean

Float

Example

```
package com.object;

public class WrapperDemo {

    public static void main(String[] args) {

        int a = 20; // primitive data type

        Integer i = new Integer(a); // autoboxing
        System.out.println("i>>" + i);

        int b = i.intValue();   //unboxing
        System.out.println("b>>"+b);
    }
}
```

Final Keyword in java

We can apply final to variables, method and class

Final variable-

A variable which is declared with final keyword called as final variables.

Once you assigned any value to that variable then it won't be changed. It works like constants in java.

How to declare the final variables-

final int a=5;

```
package com.velocity;

class FinalDemo {
public static void main(String args[]){
final int a=5;
System.out.println(a);
}
}
```

Example 2

```
package com.velocity;

class FinalDemo{
public static void main(String args[]){
 final int a=5;
 for (int a=5; a<=10;a++){
   System.out.println(a);
   }
 }
}
```

In this example, we will get compile time error, final variable values does not changed.

2.Final method-

Method which is defined with final keyword called as final method.

Note- Final method cannot be overridden.

How to declare the final method-

public final void test(){

//business logic here.

}

```
package com.velocity;

class X{
final void test(){
System.out.println("This is x class-test method");
}

class Y extends X{
final void test(){
System.out.println("This is y class-test method");
}

public static void main(String args[]){
 X x= new Y();
 x.test();
}
}
```

In this example, we will get compile time error final method cannot be override final method from X

3.Final class-

The class which is defined with final keyword called as final class.

How to declare the final class

final class Test{

//business logic

}

How you stop others from inheriting your class?

> By making class as final.

```java
public final class Main {

    public final void test() {
        System.out.println("Main class-test method");
    }

}
```

```java
package com.velocity.abstractclass;

public class Impl extends Main {

}
```

The type Impl cannot subclass the final class Main

1 quick fix available:

Remove 'final' modifier of 'Main'