

Object class

- Object class is defined in java. lang package
- Object class is a supermost parent class for all the classes in java.
- In Object class there are 11 non-static methods.

Methods in Object class:

public String toString()

public boolean equals(Object o)

public int hashCode()

protected Object clone()

protected void finalize()

final public void wait()

final public void wait(long l)

final public void wait(long l,int i)

final public void notify()

final public void notifyAll()

final public void getClass()

toString() Method

- toString() method returns String
- toString() implementation of Object class returns the reference of an Object in the String format.

Return Format:Classname@Hexadecimal

Example:

```
class Demo
{
    public static void main(String[] args)
    {
        Demo d=new Demo();
        System.out.println(d);//Demo@219828214
    }
}
```

Purpose of overriding toString():

We override toString() method to return a value based on state of the Object

Example

```
class ToStringDemo
{
    String bname;
    int bid;
    ToStringDemo(String bname,int bid)
    {
        this.bname=bname;
        this.bid=bid;
    }
    public String toString()
    {
        return "bname is " +bname+ " " + "bid is "+bid;
    }
    public static void main(String[] args)
    {
        ToStringDemo d=new ToStringDemo("java",123);
        System.out.println(d);//java 123
    }
}
```

equals(Object) Method

- The return type of equals(Object) method is boolean.
- To equals(object) method we can pass reference of any object.
- The java.lang.Object class implementation of equals(Object) method is used to compare the reference of two objects
- **Example:**

```
class Book
```

```
{ String bname;
```

```
    int bid;
```

```
Book(String bname, int bid)
```

```
{          this.bname=bname;
```

```
          this.bid=bid;
```

```
}
```

```
public boolean equals(Object o)
```

```
{
```

```
    Book b=(Book)o;
```

```
    return this.bname.equals(b.bname) && this.bid==b.bid;
```

```
}
```

```
}
```

Purpose of equals(object)

- We can override to equals(object) method to compare the state of two objects instead of comparing reference of two objects.

Note:

- If equals(object) method is not overridden it compares the reference of two objects similar to == operator
- If equals(object) method is overridden it compares the state of objects, in such case comparing the reference of the two objects is possible only by == operator

Design tip:

In equals method compare the state of an current(this) object with the passed object by downcasting the passed object.

hashCode() Method

- The return type of hashCode() method is int.
- The java.lang.Object implementation of hashCode() method is used to give the unique integer number for every object created
- The unique number is generated based on the reference of an Object.

Purpose of Overriding hashCode()

- We can override hashCode() method to return a unique integer based on the state of two objects instead based on reference of two objects.
- If equals(object) method is overridden, then it is necessary to override the hashCode() method.

Design tip:

hashCode() method should return an number based on the state of object

Example

```
class Pen
{
    double price;
    Pen(double price)
    {
        this.price=price;
    }
    public int hashCode()
    {
        int hc=(int)price;
        return hc;
    }
}
```