Object class

Object class is the predefined class in java which acts as super most class for all the classes.
Object class belongs to java.lang package.
Fully Qualified name for Object class is java.lang.Object

Methods of Object class

1. toString()
2. equals(Object obj)
3. hashCose()

1.toString()
it is a method of object class
when ever we print object reference it will call toString() method implicitly and provides complete information of an object in Hexa String format.

Complete information consists of
Packagename.classname@ObjectAddress
ex:- javaPrograms.Employee@cac736f

Syntax:-

```
public String toString() {
        //return packagename.classname@objectaddress;
}
```

Before compilation

```
package javaPrograms;

public class Employee {
        public static void main(String[] args) {
                Employee e=new Employee();
                System.out.println(e);
        }
}
```

After Compilation

```
package javaPrograms;

public class Employee {
        public static void main(String[] args) {
                Employee e=new Employee();
                System.out.println(e.toString());
        }
}
```

Purpose of overriding toString() :-

Object class implementation of toString() returns reference of the object in String format, instead of address if we want to get states (or) properties of the Object we should override toString().

ex:-

```
public class Employee {
	int eno;
	String name;
	Employee(int eno, String name){
		this.name=name;
		this.eno=eno;
	}
	public String toString() {
		return "Employee [eno=" + eno + ", name=" + name + "]";
	}
	public static void main(String[] args) {
		Employee e=new Employee(123,"joseph");
		System.out.println(e);
	}
}
```

2. equals()

it's a method of object class it compares two objects based on object address.
if object address is same, output is true else it is false.

Syntax:-

```
public boolean equals(Object o){
	return;
}
```

ex:-

```
public class Employee{
	public static void main(String[] args) {
		Employee e=new Employee();
		System.out.println(e);
		Employee e1=new Employee();
		System.out.println(e1);
		System.out.println(e.equals(e1)); //false
		e=e1;
		System.out.println(e.equals(e1)); //true
	}
}
```

Purpose of overriding equals() :-

Object class implementation of equals() compares two objects references, instead of this if we want to compare states of the object we should override equals().

ex:-

```
public class Employee {
	int eno;
	String name;
	Employee(int eno, String name){
		this.name=name;
		this.eno=eno;
	}
	public String toString() {
		return "Employee [eno=" + eno + ", name=" + name + "]";
	}
	public boolean equals(Object obj) {
		Employee other = (Employee) obj;
		return eno == other.eno && name.equals(other.name);
	}
	public static void main(String[] args) {
		Employee e=new Employee(123,"joseph");
		System.out.println(e);
		Employee e1=new Employee(123,"joseph");
		System.out.println(e1);
		Employee e2=new Employee(321,"jhonny");
		System.out.println(e2);
		System.out.println(e.equals(e1));
		System.out.println(e.equals(e2));
	}
}
```

3. hashCode() :-

   it is a method of object class, it will return the address of object in integer form.

   Contract between equals() & hashCode()

   If equals method returns true then hashCode() should give same integer for two object reference.

   If equals method returns false then hashCode() should give different integer for two object reference.

ex:-

```
public class Employee {
	int eno;
	String name;
	Employee(int eno, String name){
		this.name=name;
		this.eno=eno;
```

```
	}
	public static void main(String[] args) {
		Employee e=new Employee(123,"joseph");
		System.out.println(e.hashCode()); //212628335
		Employee e1=new Employee(123,"joseph");
		System.out.println(e1.hashCode()); //1579572132
		Employee e2=new Employee(321,"jhonny");
		System.out.println(e2.hashCode()); //359023572
	}
}
```

Purpose of overriding hashCode() :-

To maintain above mentioned contract between hashCode() & equals().

ex:-

```
public class Employee {
	int eno;
	String name;
	Employee(int eno, String name){
		this.name=name;
		this.eno=eno;
	}
	@Override
	public String toString() {
		return "Employee [eno=" + eno + ", name=" + name + "]";
	}
	@Override
	public int hashCode() {
		return Objects.hash(eno, name);
	}
	@Override
	public boolean equals(Object obj) {
		Employee other = (Employee) obj;
		return eno == other.eno && name.equals(other.name);
	}
	public static void main(String[] args) {
		Employee e=new Employee(123,"joseph");
		System.out.println(e.hashCode()); //-1154235083
		Employee e1=new Employee(123,"joseph");
		System.out.println(e1.hashCode()); //-1154235083
		Employee e2=new Employee(321,"jhonny");
		System.out.println(e2.hashCode()); //-1160804152
	}
}
```