General Contract of equals() method

There are some general principles defined by [Java SE](https://www.javatpoint.com/java-se) that must be followed while implementing the equals() method in Java. The equals() method must be:

* ***reflexive***: An object x must be equal to itself, which means, for object x, **equals(x)** should return true.
* ***symmetric***: for two given objects x and y, *x.equals(y)* must return true if and only if *equals(x)*returns true.
* ***transitive***: for any objects x, y, and z, if x.equals(y) returns true and y.equals(z) returns true, then x.equals(z) should return true.
* ***consistent***: for any objects x and y, the value of x.equals(y) should change, only if the property in equals() changes.
* For any object x, the *equals(null)* must return false.

Java hashcode()

* A **hashcode** is an integer value associated with every object in Java, facilitating the hashing in hash tables.
* To get this hashcode value for an object, we can use the hashcode() method in Java. It is the means ***hashcode() method that returns the integer hashcode value of the given object***.
* Since this method is defined in the Object class, hence it is inherited by user-defined classes also.
* The hashcode() method returns the same hash value when called on two objects, which are equal according to the equals() method. And if the objects are unequal, it usually returns different hash values.

Syntax:

1. **public** **int** hashCode()

Returns:

It returns the hash code value for the given objects.

Contract for hashcode() method in Java

* If two objects are the same as per the equals(Object) method, then if we call the hashCode() method on each of the two objects, it must provide the same integer result.

