. equals() method .

In java equals() method is used to compare equality of two Objects. The equality can be compared in two ways:
Shallow comparison: The default implementation of equals method is defined in Java.lang.Object class which simply checks if two Object references (say x and y) refer to the same Object. i.e. It checks if x == y. Since the Object class has no data members that define its state, it is also known as shallow comparison.
Deep Comparison: Suppose a class provides its own implementation of equals() method in order to compare the Objects of that class w.r.t state of the Objects. That means data members (i.e. fields) of Objects are to be compared with one another. Such Comparison based on data members is known as deep comparison.

Syntax : public boolean equals (Object obj)

General Contract of equals() method

- Reflexive , Symmetric , Transitive ,Consistent

Note: For any non-null reference value a, a.equals(null) should return false.

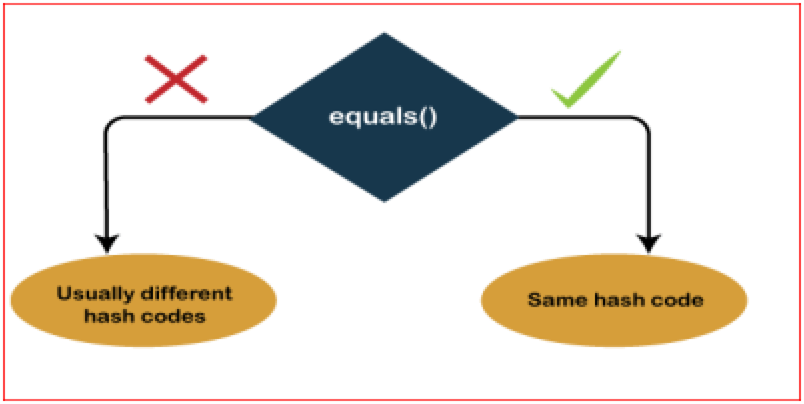
Contributed by: Mohd ShahMukit

. hashCode() method .

It returns the hashcode value as an Integer. Hashcode value is mostly used in hashing based collections like HashMap, HashSet, HashTable....etc. This method must be overridden in every class which overrides equals() method.

Syntax : public int hashCode()

Contract of hashcode() method
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.



Note: Equal objects must produce the same hash code as long as they are equal, however unequal objects need not produce distinct hash codes.