**equals() and hashCode() Contract**

equals() and hashCode() methods provided by java.lang.Object class.

**equals()**

This method checks for content equality. The two objects are equal if they store at the same memory location.

**Syntax:**

public boolean equals(Object obj)

**equals method contract**

There are some rules defined by JAVA SE that must be followed by the equals () method as shown below:

* **Reflexive:** Object will always be equal to itself as a.equals(a) will always return true.
* **Symmetric:** Object a and b, if a.equals(b) returns true then b.equals(a) will always return true
* **Transitive:** Object a, b, and c, if a.equals(b) returns true and b.equals(c) returns true, then a.equals(c) will always return true.
* **Consistent:** Object a and b, checking multiple times a.equals(b) will always return the same value, either true or false

**hashCode()**

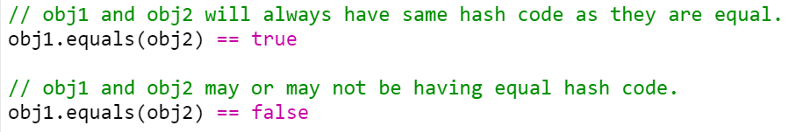
This method returns an integer value. This method is used to compare objects and needs to be overridden in every class which overrides the equals() method.

**Syntax**

public int hashCode()

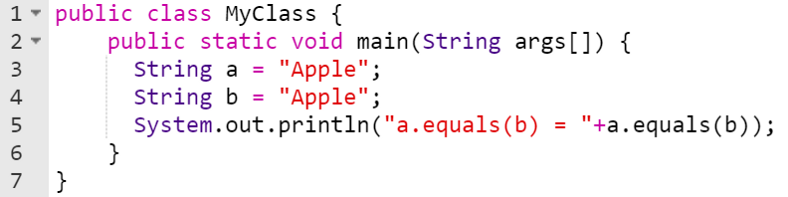
hashCode() method internally calculates the hashCode of the current Object and returns int value.

**NOTE:** Objects that are equal will always be having same hash code. Vice versa is not correct, two unequal Objects can have same or different hashcode.



**Example**

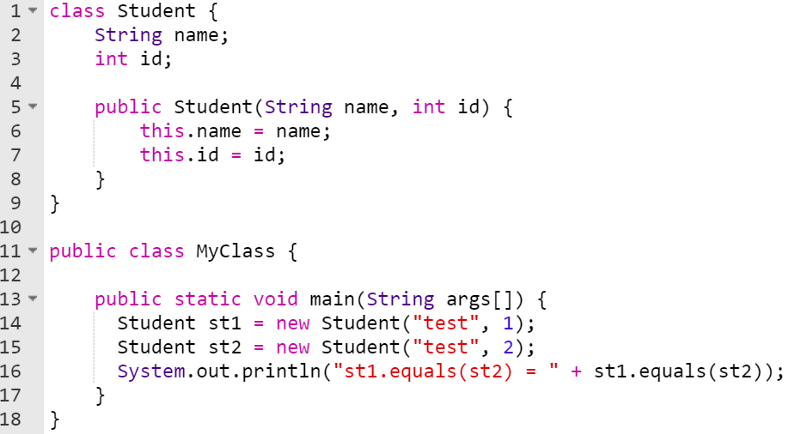
In below example two different variables a and b are having same String value as Apple. Checking equality on line number 5 **“a.equals(b)”** returns true as both the String got matched.



OUTPUT:

a.equals(b) = true

Now, in below example we created two different objects with same values at line number 14 and 15. While checking equality at line number 16 it returned false.



**OUTPUT:**

st1.equals(st2) = false

**hashCode() and equals() contract example**

In below example, we have proved **st1.equals(st2) as true** after overriding equals() and hashCode() methods.

st1.equals(st2) = true

In below example we have overridden the equals method to check object equality.

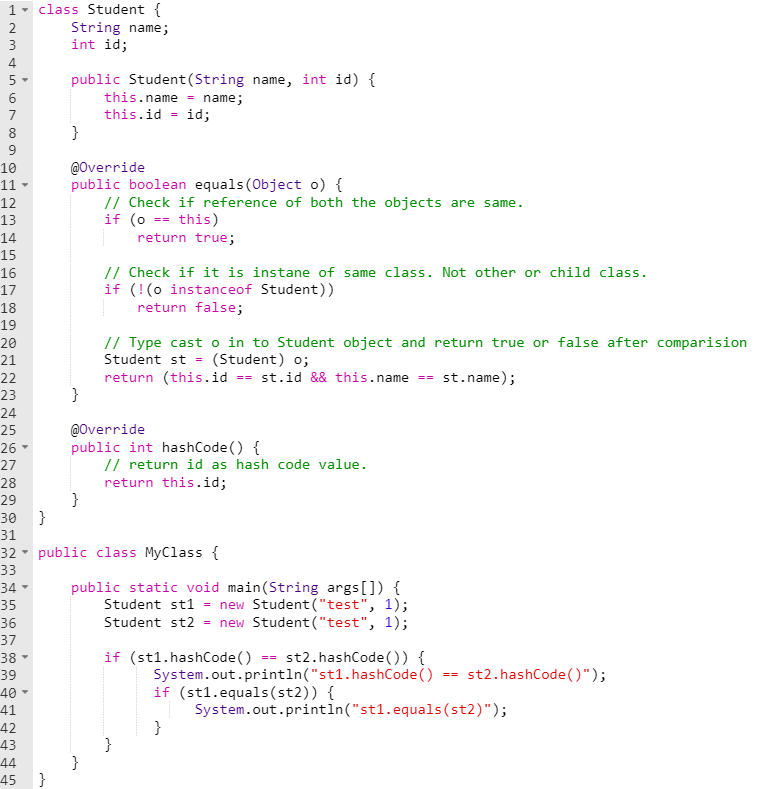
equals() method override:

On line number 13, it checks if two objects have the same reference. If yes, they are referring to the same object and this means, they are equal by default.

On line number 17, it checks if an object belongs to the same class or not. There is no need to compare if the object is not an instance of the required class. In this case comparison object is incorrect and will always be unequal.

Finally, on line number 22, current object compare field value with object passed in a parameter. Objects will be unequal by default if any of the field value didn’t get match..

According to Java SE, two objects are equal if and only if their hash codes are also equal. To demonstrate this condition, the hashCode () method must be overridden. On line number 25, after overriding the hashCode() method, it is returning id as a hashcode as this will satisfy the condition.



OUTPUT:



I hope you found out this article interesting and informative. Please share it with your friends to spread the knowledge.