

hashCode() in java

```
class Contact {  
    private int contactId;  
    private String contactName;  
    private String mobileNumber;  
    Contact(int contactId, String contactName, String mobileNumber) {  
        this.contactId = contactId;  
        this.contactName = contactName;  
        this.mobileNumber = mobileNumber;  
    }  
}  
  
public class ContactsChecker {  
    public static void main(String arg[]) {  
        Contact contact1 = new Contact( contactId: 1, contactName: "Priya", mobileNumber: "6382709056");  
        Contact contact2 = new Contact( contactId: 1, contactName: "Priya", mobileNumber: "6382709056");  
        Contact contact3 = new Contact( contactId: 2, contactName: "Nithya", mobileNumber: "9876543210");  
        Contact contact4 = contact3;  
        System.out.println(contact1.hashCode());  
        System.out.println(contact2.hashCode());  
        System.out.println(contact3.hashCode());  
        System.out.println(contact4.hashCode());  
    }  
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...  
2133927002  
1836019240  
325040804  
325040804
```

In Java, hashCode() methods returns the hash code of the respective object.

The hash code of the objects will be same when it is copied but won't be same when it has same values.

"==" in Java

```
class Contact {
    private int contactId;
    private String contactName;
    private String mobileNumber;
    Contact(int contactId, String contactName, String mobileNumber) {
        this.contactId = contactId;
        this.contactName = contactName;
        this.mobileNumber = mobileNumber;
    }
}

public class ContactsChecker {
    public static void main(String arg[]) {
        Contact contact1 = new Contact( contactId: 1, contactName: "Priya", mobileNumber: "6382709056");
        Contact contact2 = new Contact( contactId: 1, contactName: "Priya", mobileNumber: "6382709056");
        Contact contact3 = new Contact( contactId: 2, contactName: "Nithya", mobileNumber: "9876543210");
        Contact contact4 = contact3; // copying the object
        System.out.println(contact1 == contact2);
        System.out.println(contact3 == contact4);
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
false
true
```

"==" operator compares reference or memory location of objects in a heap, whether they point to the same location or not.

equals() in Java

```
class Contact {
    private int contactId;
    private String contactName;
    private String mobileNumber;

    Contact(int contactId, String contactName, String mobileNumber) {
        this.contactId = contactId;
        this.contactName = contactName;
        this.mobileNumber = mobileNumber;
    }
}

public class ContactsChecker {
    public static void main(String arg[]) {
        Contact contact1 = new Contact(1, "Priya", "6382709056");
        Contact contact2 = new Contact(1, "Priya", "6382709056");
        Contact contact3 = new Contact(2, "Nithya", "9876543210");
        Contact contact4 = contact3; // copying the object
        System.out.println(contact1.equals(contact2));
        System.out.println(contact3.equals(contact4));
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
false
true
```

equals() method checks whether the objects has same hash code by calling hashCode() method. Since contact1 and contact2 have different hash code it returns false.

How to Overcome this issue in Java?

By overriding the equals() and hashCode() methods In Java, we can over come this issue.

```
@Override
public boolean equals(Object obj)
{
    if(this == obj) // if both are same reference
        return true;
    if(!(obj instanceof Contact)) // whether the object belong to this class
        return false;
    Contact contact = (Contact) obj; //typecasting the object
    return (contact.contactName.equals(this.contactName) &&
            contact.contactId == this.contactId &&
            contact.mobileNumber.equals(this.mobileNumber));
}
```

```
@Override
public int hashCode()
{
    return Objects.hash(contactName,contactId, mobileNumber);
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
true
true
-798067691|
-798067691
```

equals() and hashCode() methods now checks the individual attributes of the object.

hashCode() in kotlin

```
class Contact(val contactId: Int, val contactName: String, val mobileNumber: String)

fun main() {
    val contact1 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact2 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact3 = Contact(contactId: 2, contactName: "Nithya", mobileNumber: "9876543210")
    val contact4 = contact3
    println(contact1.hashCode())
    println(contact2.hashCode())
    println(contact3.hashCode())
    println(contact4.hashCode())
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
2133927002
1836019240
325040804
325040804
```

is same as hashCode() in java.

"==" in kotlin

```
class Contact(val contactId: Int, val contactName: String, val mobileNumber: String)

fun main() {
    val contact1 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact2 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact3 = Contact(contactId: 2, contactName: "Nithya", mobileNumber: "9876543210")
    val contact4 = contact3
    println(contact1 == contact2)
    println(contact3 == contact4)
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
false
true
```

In Kotlin, == is the default way to compare two objects: it compares their values by calling equals under the hood.

equals() in kotlin

```
class Contact(val contactId: Int, val contactName: String, val mobileNumber: String)

fun main() {
    val contact1 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact2 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact3 = Contact(contactId: 2, contactName: "Nithya", mobileNumber: "9876543210")
    val contact4 = contact3
    println(contact1.equals(contact2))
    println(contact3.equals(contact4))
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
false
true
```

How to Overcome this issue in Kotlin?

by using data class

```
data class Contact(val contactId: Int, val contactName: String, val mobileNumber: String)

fun main() {
    val contact1 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact2 = Contact(contactId: 1, contactName: "Priya", mobileNumber: "6382709056")
    val contact3 = Contact(contactId: 2, contactName: "Nithya", mobileNumber: "9876543210")
    val contact4 = contact3
    println(contact1 == contact2)
    println(contact1.hashCode())
    println(contact2.hashCode())
}
```

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
true
250522970
250522970
```

In data class the hashCode() for the equal objects are same. So the equals method returns true.