**What is the need to Override Hashcode() and equals() method**

http://0.gravatar.com/avatar/30e112f7dd512c86ac3e25f10d669bd0?s=30&d=mm&r=gAuthor posted by [Jitendra](http://www.jitendrazaa.com/blog/author/admin/) on Posted on[April 11, 2011](http://www.jitendrazaa.com/blog/java/what-is-the-need-to-override-hashcode-and-equals-method/) under category Categories[JAVA](http://www.jitendrazaa.com/blog/category/java/) and tagged as Tags[JAVA](http://www.jitendrazaa.com/blog/tag/java/) with [36 Comments](http://www.jitendrazaa.com/blog/java/what-is-the-need-to-override-hashcode-and-equals-method/#disqus_thread)

Although there are lots of material are available on internet and API document about the necessity of the overriding the hashcode() and equals() method in Java but lots of new developers still not able to understand the necessity of hashcode() method.  
In this article, I will try to explain step by step the need of overriding hashcode() method in Java.

Although there are lots of materials are available on internet and API document about the necessity of the overriding the **hashcode()**and **equals()**method in Java but lots of new developers still not able to understand the necessity of **hashcode()**method.  
In this article, I will try to explain step by step the need of overriding **hashcode()** method in Java.

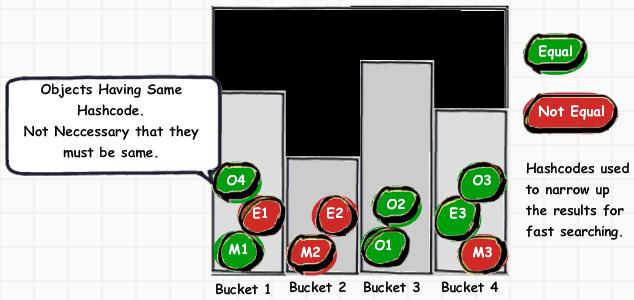
**Few Thump rules:**

* If two objects are same then they must return same value in hashcode() and equals() method whenever invoked.
* It is not necessary that two different object must have different hashcode values. it might be possible that they share common hash bucket.

*JVM assigns unique hashcode value to each object when they are created in memory and if developers don’t override the hashcode method then there is no way the two object returns same hashcode value.*

As the question comes in your mind that equals() method is used to compare objects that they are having same value or not but **why should we override the hashcode method ?**

The answer to the question is for the hash technique based data structures like HashMap and HashTable.

[](http://jitendrazaa.com/blog/wp-content/uploads/2011/04/How-Hashcode-works-in-java.jpg)How Hashcode works in java

As you can see in above diagram that every object is placed in Hash bucket depending on the hashcode they have. It is not necessary that every different object must have different hashcode. **hashcode is used to narrow the search result.**When we try to insert any key in HashMap first it checks whether any other object present with same hashcode and if yes then it checks for the equals() method. If two objects are same then HashMap will not add that key instead it will replace the old value by new one.

**What will happen if I don’t override the hashcode method?**  
Ans : If the object does not implement hashcode() method and used as key then we will not get the object back as shown in below code.

**Code without implementation of equals() and hashcode()**

|  |  |  |
| --- | --- | --- |
| 1 | package com.G2.Collections; | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | import java.util.HashMap; | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | class Movie { |
| 6 | private String name, actor; | |

|  |  |
| --- | --- |
| 7 |  |
| 8 | public String getName() { | |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | return name; | | |
| 10 | | } |

|  |  |
| --- | --- |
| 11 |  |
| 12 | public void setName(String name) { | |

|  |  |  |
| --- | --- | --- |
| 13 | this.name = name; | |
| 14 | } |

|  |  |
| --- | --- |
| 15 |  |
| 16 | public String getActor() { | |

|  |  |  |
| --- | --- | --- |
| 17 | return actor; | |
| 18 | } |

|  |  |
| --- | --- |
| 19 |  |
| 20 | public void setActor(String actor) { | |

|  |  |  |
| --- | --- | --- |
| 21 | this.actor = actor; | |
| 22 | } |

|  |  |
| --- | --- |
| 23 |  |
| 24 | public int getReleaseYr() { | |

|  |  |  |
| --- | --- | --- |
| 25 | return releaseYr; | |
| 26 | } |

|  |  |
| --- | --- |
| 27 |  |
| 28 | public void setReleaseYr(int releaseYr) { | |

|  |  |  |
| --- | --- | --- |
| 29 | this.releaseYr = releaseYr; | |
| 30 | } |

|  |  |
| --- | --- |
| 31 |  |
| 32 | private int releaseYr; | |

|  |  |  |
| --- | --- | --- |
| 33 | } | |
| 34 |  |

|  |  |  |
| --- | --- | --- |
| 35 | public class HashMapDemo { | |
| 36 |  |

|  |  |  |
| --- | --- | --- |
| 37 | public static void main(String[] args) { | |
| 38 |  |

|  |  |  |
| --- | --- | --- |
| 39 | Movie m = new Movie(); | |
| 40 | m.setActor("Akshay"); |

|  |  |  |
| --- | --- | --- |
| 41 | m.setName("Thank You"); | |
| 42 | m.setReleaseYr(2011); |

|  |  |
| --- | --- |
| 43 |  |
| 44 | Movie m1 = new Movie(); | |

|  |  |
| --- | --- |
| 45 | m1.setActor("Akshay"); |
| 46 | m1.setName("Khiladi"); |

|  |  |  |
| --- | --- | --- |
| 47 | m1.setReleaseYr(1993); | |
| 48 |  |

|  |  |  |
| --- | --- | --- |
| 49 | Movie m2 = new Movie(); | |
| 50 | m2.setActor("Akshay"); |

|  |  |
| --- | --- |
| 51 | m2.setName("Taskvir"); |
| 52 | m2.setReleaseYr(2010); |

|  |  |
| --- | --- |
| 53 |  |
| 54 | Movie m3 = new Movie(); | |

|  |  |
| --- | --- |
| 55 | m3.setActor("Akshay"); |
| 56 | m3.setName("Taskvir"); |

|  |  |  |
| --- | --- | --- |
| 57 | m3.setReleaseYr(2010); | |
| 58 |  |

|  |  |  |
| --- | --- | --- |
| 59 | HashMap<Movie, String> map = new HashMap<Movie, String>(); | |
| 60 | map.put(m, "ThankYou"); |

|  |  |  |
| --- | --- | --- |
| 61 | map.put(m1, "Khiladi"); | |
| 62 | map.put(m2, "Tasvir"); |

|  |  |  |
| --- | --- | --- |
| 63 | map.put(m3, "Duplicate Tasvir"); | |
| 64 |  |

|  |  |
| --- | --- |
| 65 | //Iterate over HashMap |
| 66 | for (Movie mm : map.keySet()) { | |

|  |  |  |
| --- | --- | --- |
| 67 | System.out.println(map.get(mm).toString()); | |
| 68 | } |

|  |  |
| --- | --- |
| 69 |  |
| 70 | Movie m4 = new Movie(); | |

|  |  |
| --- | --- |
| 71 | m4.setActor("Akshay"); |
| 72 | m4.setName("Taskvir"); |

|  |  |  |
| --- | --- | --- |
| 73 | m4.setReleaseYr(2010); | |
| 74 |  |

|  |  |  |
| --- | --- | --- |
| 75 | /\* We are trying to retrieve m2, by creating object m4 with exact values as of m2, However Hashcode method is not implemented and that why we are not able to get Object m2 \*/ | |
| 76 | if(map.get(m4) == null ){ |

|  |  |
| --- | --- |
| 77 | System.out.println("----------------"); |
| 78 | System.out.println("Object not found"); |

|  |  |  |
| --- | --- | --- |
| 79 | System.out.println("----------------"); | |
| 80 | }else{ |

|  |  |  |
| --- | --- | --- |
| 81 | System.out.println(map.get(m4).toString()); | |
| 82 | } |

|  |  |  |
| --- | --- | --- |
| 83 | } | |
| 84 | } |

**Output:**  
Khiladi  
Tasvir  
ThankYou  
Duplicate Tasvir  
—————-  
Object not found  
—————-

As you can see in above program :

1. Duplicate objects are added in Hashmap as a key (Because we have not overided the hashcode and equals method)
2. We are not able to get back object from map (Because hashcode is not implemented)

**Same program with equals and hashcode implementation:**

|  |  |  |
| --- | --- | --- |
| 1 | package com.G2.Collections; | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | import java.util.HashMap; | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | class Movie { |
| 6 | private String name, actor; | |

|  |  |
| --- | --- |
| 7 |  |
| 8 | @Override | |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | public boolean equals(Object o) { | | |
| 10 | | Movie m = (Movie) o; |

|  |  |  |
| --- | --- | --- |
| 11 | return m.actor.equals(this.actor) && m.name.equals(this.name) && m.releaseYr == this.releaseYr; | |
| 12 | } |

|  |  |
| --- | --- |
| 13 |  |
| 14 | @Override | |

|  |  |
| --- | --- |
| 15 | public int hashCode() { |
| 16 | return actor.hashCode() + name.hashCode() + releaseYr; | |

|  |  |  |
| --- | --- | --- |
| 17 | } | |
| 18 |  |

|  |  |  |
| --- | --- | --- |
| 19 | public String getName() { | |
| 20 | return name; |

|  |  |  |
| --- | --- | --- |
| 21 | } | |
| 22 |  |

|  |  |  |
| --- | --- | --- |
| 23 | public void setName(String name) { | |
| 24 | this.name = name; |

|  |  |  |
| --- | --- | --- |
| 25 | } | |
| 26 |  |

|  |  |  |
| --- | --- | --- |
| 27 | public String getActor() { | |
| 28 | return actor; |

|  |  |  |
| --- | --- | --- |
| 29 | } | |
| 30 |  |

|  |  |  |
| --- | --- | --- |
| 31 | public void setActor(String actor) { | |
| 32 | this.actor = actor; |

|  |  |  |
| --- | --- | --- |
| 33 | } | |
| 34 |  |

|  |  |  |
| --- | --- | --- |
| 35 | public int getReleaseYr() { | |
| 36 | return releaseYr; |

|  |  |  |
| --- | --- | --- |
| 37 | } | |
| 38 |  |

|  |  |  |
| --- | --- | --- |
| 39 | public void setReleaseYr(int releaseYr) { | |
| 40 | this.releaseYr = releaseYr; |

|  |  |  |
| --- | --- | --- |
| 41 | } | |
| 42 |  |

|  |  |  |
| --- | --- | --- |
| 43 | private int releaseYr; | |
| 44 | } |

|  |  |
| --- | --- |
| 45 |  |
| 46 | public class HashMapDemo { | |

|  |  |
| --- | --- |
| 47 |  |
| 48 | public static void main(String[] args) { | |

|  |  |
| --- | --- |
| 49 |  |
| 50 | Movie m = new Movie(); | |

|  |  |
| --- | --- |
| 51 | m.setActor("Akshay"); |
| 52 | m.setName("Thank You"); | |

|  |  |  |
| --- | --- | --- |
| 53 | m.setReleaseYr(2011); | |
| 54 |  |

|  |  |  |
| --- | --- | --- |
| 55 | Movie m1 = new Movie(); | |
| 56 | m1.setActor("Akshay"); |

|  |  |
| --- | --- |
| 57 | m1.setName("Khiladi"); |
| 58 | m1.setReleaseYr(1993); |

|  |  |
| --- | --- |
| 59 |  |
| 60 | Movie m2 = new Movie(); | |

|  |  |
| --- | --- |
| 61 | m2.setActor("Akshay"); |
| 62 | m2.setName("Taskvir"); |

|  |  |  |
| --- | --- | --- |
| 63 | m2.setReleaseYr(2010); | |
| 64 |  |

|  |  |  |
| --- | --- | --- |
| 65 | Movie m3 = new Movie(); | |
| 66 | m3.setActor("Akshay"); |

|  |  |
| --- | --- |
| 67 | m3.setName("Taskvir"); |
| 68 | m3.setReleaseYr(2010); |

|  |  |
| --- | --- |
| 69 |  |
| 70 | HashMap<Movie, String> map = new HashMap<Movie, String>(); | |

|  |  |
| --- | --- |
| 71 | map.put(m, "ThankYou"); |
| 72 | map.put(m1, "Khiladi"); |

|  |  |
| --- | --- |
| 73 | map.put(m2, "Tasvir"); |
| 74 | map.put(m3, "Duplicate Tasvir"); | |

|  |  |
| --- | --- |
| 75 |  |
| 76 | // Iterate over HashMap | |

|  |  |
| --- | --- |
| 77 | for (Movie mm : map.keySet()) { |
| 78 | System.out.println(map.get(mm).toString()); | |

|  |  |  |
| --- | --- | --- |
| 79 | } | |
| 80 |  |

|  |  |  |
| --- | --- | --- |
| 81 | Movie m4 = new Movie(); | |
| 82 | m4.setActor("Akshay"); |

|  |  |
| --- | --- |
| 83 | m4.setName("Taskvir"); |
| 84 | m4.setReleaseYr(2010); |

|  |  |
| --- | --- |
| 85 |  |
| 86 | if (map.get(m4) == null) { | |

|  |  |
| --- | --- |
| 87 | System.out.println("----------------"); |
| 88 | System.out.println("Object not found"); |

|  |  |  |
| --- | --- | --- |
| 89 | System.out.println("----------------"); | |
| 90 | } else { |

|  |  |
| --- | --- |
| 91 | System.out.println("----------------"); |
| 92 | System.out.println(map.get(m4).toString()); | |

|  |  |  |
| --- | --- | --- |
| 93 | System.out.println("----------------"); | |
| 94 | } |

|  |  |  |
| --- | --- | --- |
| 95 | } | |
| 96 | } |

**Output:**  
Khiladi  
Duplicate Tasvir  
ThankYou  
—————-  
Duplicate Tasvir  
—————-  
As you can see :

* Duplicate Keys are not added instead there values are replaced.
* Now the object is retrieved from the Map.