# The hashCode() method of java.lang.Object

Anytime you override equals() you should also override hashCode(). The hashCode() method should ideally return the same int for any two objects that compare equal and a different int for any two objects that don't compare equal, where equality is defined by the equals() method. This is used as an index by the java.util.Hashtable class.

In the Car example equality is determined exclusively by comparing license plates; therefore only the licesePlate field is used to determine the hash code. Since licensePlate is a String, and since the String class has its own hashCode() method, we can sponge off of that.

public int hashCode() {  
   
 return this.licensePlate.hashCode();  
   
 }

Other times you may need to use the bitwise operators to merge hash codes for multiple fields. There are also a variety of useful methods in the type wrapper classes (java.lang.Double, java.lang.Float, etc.) that convert primitive data types to integers that share the same bit string. These can be used to hash primitive data types.

[Previous](http://docs.google.com/37.html) | [Next](http://docs.google.com/39.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997, 1998, 2001 Elliotte Rusty Harold

[elharo@metalab.unc.edu](mailto:elharo@metalab.unc.edu)

Last Modified November 2, 2001