*// Equality can be checked with only same class or parent-child relationship classes*

Java:

**package** com.example.nav\_contacts.presentation.activity;  
  
**import** java.util.Objects;  
  
*// equality checks:*  
 *// \* referential check*  
 *// \* values check*  
**class** Ticket{  
 **private int ticketId**;  
 String **movieName**;  
 String **showTime**;  
 String **seatNo**;  
 Ticket(**int** id, String name, String showTime, String seatNo){  
 **this**.**ticketId** = id;  
 **this**.**movieName** = name;  
 **this**.**showTime** = showTime;  
 **this**.**seatNo** = seatNo;  
 }  
 @Override  
 **public boolean** equals(Object object) {  
 *// if both are same reference*  
**if**(**this** == object) {  
 **return true**;  
 }  
 *// whether the object belong to this class*  
**if**(!(object **instanceof** Ticket))  
 **return false**;  
 Ticket ticket = (Ticket) object;  
 **return** (ticket.**movieName**.equals(**this**.**movieName**) && ticket.**ticketId** == **this**.**ticketId** &&  
 ticket.**showTime**.equals(**this**.**showTime**) && ticket.**seatNo**.equals(**this**.**seatNo**));  
 }  
 @Override  
 **public int** hashCode()  
 {  
 **return** Objects.*hash*(**ticketId**, **movieName**, **showTime**, **seatNo**);  
 }  
}

**public class** EqualityCheck {  
*// In primitives(int, char, float, double) '==' checks value*  
**void** main() {  
 **int** a = 10;  
 **int** b = 20;  
 **int** c = 30;  
 System.***out***.println(a == b); *//true*  
  
*// In Objects '==' checks reference*  
String s = **"nav"**;  
 String s2 = **"nav"**;  
 String s3 = **"vis"**;  
 System.***out***.println( s == s2); *// true*  
System.***out***.println( s2 == s3); *// false*  
  
Ticket t1 = **new** Ticket(1, **"KGF"**, **"11.30am"**, **"F3"**);  
 Ticket t2 = **new** Ticket(1, **"KGF"**, **"11.30am"**, **"F3"**);  
 Ticket t3 = **new** Ticket(3, **"KGF-2"**, **"09.30am"**, **"F9"**);  
  
 System.***out***.println(t1==t2); *//false*  
System.***out***.println( t3 == t2); *// false*  
  
*// using 'equals' without overriding methods equals and hashCode*  
System.***out***.println(t1.equals(t2)); *// false*  
System.***out***.println(t2.equals(t3)); *// false*  
  
*// using 'equals' after overriding methods equals and hashCode*  
*// (overriding any one methos voilates the Object Contract)*  
 *// hash codes*  
System.***out***.println(t1.hashCode()); *// 2463536537*  
System.***out***.println(t2.hashCode()); *// 2463536537*  
System.***out***.println(t3.hashCode()); *// 73463634634*  
  
System.***out***.println(t1.equals(t2)); *// true*  
System.***out***.println(t2.equals(t3)); *// false*  
}  
}

Kotlin:

*//== compares the value for primitive and dataclass and reference for normal class*  
*//equals compares the value*  
*//=== compares the reference*

**package** com.example.nav\_contacts.presentation.activity  
  
**class** Ticketm(  
 **private var ticketId**: Int = 0,  
 **var movieName**: String? = **null**,  
 **var showTime**: String? = **null**,  
 **var seatNo**: String? = **null**) {  
}  
**class** EqualityCheck {  
 **fun** main() {  
 **var** t1 = Ticketm(1, **"KGF"**, **"11.30am"**, **"F3"**)  
 **var** t2 = Ticketm(1, **"KGF"**, **"11.30am"**, **"F3"**)  
 **var** t3 = Ticketm(3, **"KGF-2"**, **"09.30am"**, **"F9"**)  
  
 *println*(t1 == t2) *//false*  
 *println*(t2 == t3) *//false*  
 *println*(t1.equals(t2)) *//false*  
 *println*(t1.hashCode()) *// 343438683463*  
 *println*(t2.hashCode()) *// 2542645324246*  
}  
}

Kotlin:

**package** com.example.nav\_contacts.presentation.activity  
  
**data class** Tickett(  
 **private var ticketId**: Int = 0,  
 **var movieName**: String? = **null**,  
 **var showTime**: String? = **null**,  
 **var seatNo**: String? = **null**) {  
}  
**class** EqualityCheckDataClass {  
 **fun** main() {  
 **var** t1 = Tickett(1, **"KGF"**, **"11.30am"**, **"F3"**)  
 **var** t2 = Tickett(1, **"KGF"**, **"11.30am"**, **"F3"**)  
 **var** t3 = Tickett(3, **"KGF-2"**, **"09.30am"**, **"F9"**)  
  
 *println*(t1 == t2) *//true*  
 *println*(t2 == t3) *//false*  
 *println*(t1.equals(t2)) *//true*  
 *println*(t1.hashCode()) *// 83478763383*  
 *println*(t2.hashCode()) *// 83478763383*  
}  
}