In Java :

“ == ” and equals() without overriding method from Object class

```
class Student{
 String name;
int age;
Student(String name, int age){
 this.name = name;
this.age = age;
}
}


public class Main {
public static void main(String[] args) {
Student student1 = new Student("Achu",2);
Student student2 = new Student("Achu",2);
Student student3 = new Student("Achuthan",3);
Student student4 = student2;
System.out.println(student1 == student2); // false
System.out.println(student1.equals(student2)); /*false – since we doesn't override equals(), its refers only reference/ memory location */
System.out.println(student4 == student2); // true
System.out.println(student4.equals(student2)); /*true – since here same reference/ memory location */
System.out.println(student1.hashCode()); // 1555009629
System.out.println(student2.hashCode()); // 41359092
```

```
System.out.println(student3.hashCode()); // 149928006

System.out.println(student4.hashCode()); // 41359092

ArrayList listOfStudents1 = new ArrayList<Student>();

listOfStudents1.add(student1);

listOfStudents1.add(student2);

ArrayList listOfStudents2 = new ArrayList<Student>();

listOfStudents2.add(student1);

listOfStudents2.add(student4);

System.out.println(listOfStudents1 == listOfStudents2); //  false

System.out.println(listOfStudents1.equals(listOfStudents2)); // true


System.out.println(listOfStudents1.hashCode()); // 1002018296

System.out.println(listOfStudents2.hashCode()); // 1002018296


}

}
```

“ == ” and equals() with overriding equals() and hashCode() methods from Object Class.

```
class Student{

 String name;

int age;

Student(String name, int age){
```

```
 this.name = name;
this.age = age;
}
public boolean equals(Object obj){
 if(this == obj)
 return true;
if(!(obj instanceof Student))
 return false;
Student student = (Student) obj;
return (this.name.equals(student.name) &&
 this.age == student.age);
}
public int hashCode() {
return Objects.hash(name, age);
}
}

public class Main {
public static void main(String[] args) {
Student student1 = new Student("Achu",2);
Student student2 = new Student("Achu",2);
Student student3 = new Student("Achuthan",3);
Student student4 = student2;
System.out.println(student1 == student2); // false
```

```
System.out.println(student1.equals(student2)); // true
System.out.println(student4 == student2); // true
System.out.println(student4.equals(student2)); // true
System.out.println(student1.hashCode()); // 63082708
System.out.println(student2.hashCode()); // 63082708
System.out.println(student3.hashCode()); // 12345678
System.out.println(student4.hashCode()); // 63082708
ArrayList listOfStudents1 = new ArrayList<Student>();
listOfStudents1.add(student1);
listOfStudents1.add(student2);
ArrayList listOfStudents2 = new ArrayList<Student>();
listOfStudents2.add(student1);
listOfStudents2.add(student4);
System.out.println(listOfStudents1 == listOfStudents2); // false
System.out.println(listOfStudents1.equals(listOfStudents2)); // true
System.out.println(listOfStudents1.hashCode()); // 2020678567
System.out.println(listOfStudents2.hashCode()); // 2020678567
}
}
```

In Kotlin :

“ == ” and equals() are used for values equality [operator overloading]

“ === ” is used for objects equality

Using Normal class :

```
class Student(val name : String, val age : Int)


fun main() {
val student1 = Student("Achu",2)
val student2 = Student("Achu",2)
val student3 = Student("Achuthan",3)
val student4 = student2
println(student1 == student2); // false
println(student1 ===  student2); // false
println(student1.equals(student2)); // false
println(student4 == student2); // true
println(student4.equals(student2)); // true
println(student1.hashCode()); //835648992
println(student2.hashCode()); // 1134517053
println(student3.hashCode()); // 492228202
println(student4.hashCode()); // 1134517053
val listOfStudent1 = listOf(student1, student2)
val listOfStudent2 = listOf(student2, student1)
println(listOfStudent1 == listOfStudent2) // false
println(listOfStudent2.equals(listOfStudent1)) // false
}
```

Issue : for variable student1 and student2 values are same both it shows both are not equal, to overcome this issues , we uses data class(which overrides equals() and hashCode() on its own).

Using Data Class :

```
data class Student(val name : String, val age : Int)

fun main() {

val student1 = Student("Achu",2)

val student2 = Student("Achu",2)

val student3 = Student("Achuthan",3)

val student4 = student2

println(student1 == student2); // true

println(student1 === student2); // false --  referntial check

println(student1.equals(student2)); // true

println(student4 == student2); // true

println(student4.equals(student2)); // true

println(student1.hashCode()); //63081747

println(student2.hashCode()); // 63081747

println(student3.hashCode()); // 492228202

println(student4.hashCode()); // 63081747

val listOfStudent1 = listOf(student1, student2)

val listOfStudent2 = listOf(student2, student1)

println(listOfStudent1 == listOfStudent2) // true

println(listOfStudent2.equals(listOfStudent1)) // true

}
```

References :

https://web.mit.edu/6.005/www/fa15/classes/15-equality/

https://medium.com/@ganeshraj020794/hashcode-and-equals-contract-f248a1d5cac9

https://medium.com/booking-com-development/object-equality-in-java-and-kotlin-9441f34e2163