





## Example Without Overriding:

```
java

☐ Copy  
② Edit

class Student {
    String name;
    Student(String name) {
        this.name = name;
Student s1 = new Student("Ali");
Student s2 = new Student("Ali");
System.out.println(s1.equals(s2)); // X false (different memory location)
```





## What if you want them to be compared by content (e.g., name)?

You need to **override** the equals() method:

```
goverride
public boolean equals(Object o) {
   if (this == o) return true;
   if (!(o instanceof Student)) return false;
   Student s = (Student) o;
   return name.equals(s.name);
}
```





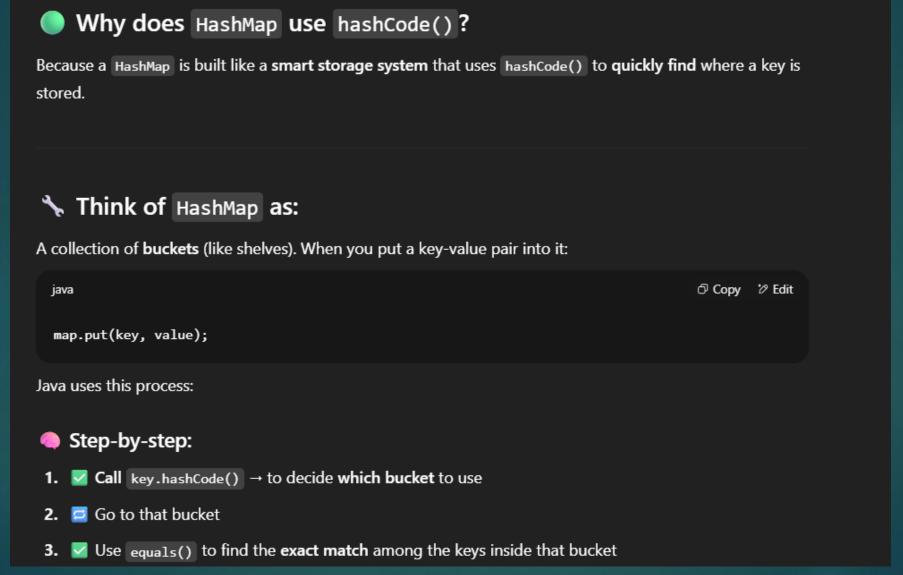
- What is hashCode() in Java?
- hashCode() returns a number (int) that represents an object.

That number is used by Java to quickly find or store objects in collections like:

- HashSet
- HashMap
- HashTable











```
☐ Copy '⊅ Edit
 text
  Buckets (array index from 0 to N):
  [0] → empty
  [1] → empty
  [2] → [ Student(name=Ali) → "Grade A" ]
  [3] → empty
When you do:

☐ Copy  
② Edit

 java
  map.get(new Student("Ali"))
Java does:
 1. Call hashCode() → gets 2, go to bucket 2
 2. Use equals() to check each key in that bucket
 3. If matches → return value
```





## If you don't override hashCode():

Even if two keys are equal by equals(), they might have different hash codes, so Java:

- Looks in the wrong bucket
- X Can't find your object





First, how do HashSet / HashMap / HashTable work?

These collections use two methods internally:

- 1. hashCode() → to find the bucket (location)
- 2. equals() → to check actual equality of objects





## Imagine:

You have a library (like a HashSet ) with thousands of books.

To find your book fast, you compute a code (like a drawer number) using hashCode().

Java uses this number to jump directly to the right "drawer" and then uses <code>.equals()</code> to check inside that drawer.





- Without hashCode():
- Java doesn't know where to look
- Even if .equals() returns true, Java might not even call it
- Result: contains() or get() fails





```
Without overriding hashCode():
                                                                                 ☐ Copy 🍪 Edit
 java
 class Student {
     String name;
     Student(String name) {
         this.name = name;
     @Override
     public boolean equals(Object o) {
        return o instanceof Student && name.equals(((Student) o).name);
     // No hashCode()
```

```
java

Student s1 = new Student("Ali");
Student s2 = new Student("Ali");

HashSet<Student> set = new HashSet<>();
set.add(s1);

// X Returns false, because hashCode is different
System.out.println(set.contains(s2));
```





Add hashCode() like this: ☐ Copy 🎖 Edit java @Override public int hashCode() { return name.hashCode(); // or Objects.hash(name); Now: ☐ Copy 🎖 Edit java System.out.println(set.contains(s2)); // // Itrue