🟦 What is `equals()` in Java?

When you compare two objects using:

```java
a.equals(b)
```

It checks: **"Are these two objects logically equal?"**

By **default**, Java compares the **memory location** (are they the same object in RAM?).

By Eslam Khder

✅ Example Without Overriding:

java

Copy Edit

```java
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)); // ❌ false (different memory location)
```

By Eslam Khder

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

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

```java
@Override
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);
}
```

spring

EraaSoft

🔷 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`

By Eslam Khder

🟢 Why does `HashMap` use `hashCode()`?

Because a `HashMap` is built like a **smart storage system** that uses `hashCode()` to **quickly find** where a key is stored.

🔧 Think of `HashMap` as:

A collection of **buckets** (like shelves). When you put a key-value pair into it:

```java
map.put(key, value);
```

Java uses this process:

🧠 Step-by-step:

1. ✅ **Call** `key.hashCode()` → to decide **which bucket** to use
2. 🔁 Go to that bucket
3. ✅ Use `equals()` to find the **exact match** among the keys inside that bucket

By Eslam Khder

text

```text
Buckets (array index from 0 to N):
[0] → empty
[1] → empty
[2] → [ Student(name=Ali) → "Grade A" ]
[3] → empty
```

When you do:

java

```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

By Eslam Khder

🔴 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**
- ❌ Can't find your object

spring

EraaSoft

🔷 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

By Eslam Khder

spring

EraaSoft

📦 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 `.equals()` to check inside that drawer.

By Eslam Khder

🔄 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

spring®

EraaSoft

Without overriding `hashCode()`:

java

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

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

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

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

By Eslam Khder

✅ Add hashCode() like this:

java

Copy Edit

```java
@Override
public int hashCode() {
    return name.hashCode(); // or Objects.hash(name);
}
```

Now:

java

Copy Edit

```java
System.out.println(set.contains(s2)); // ✅ true
```

By Eslam Khder