

## Overriding equals method in Java

Consider the following Java program:

```
class Complex {
    private double re, im;

    public Complex(double re, double im) {
        this.re = re;
        this.im = im;
    }
}

// Driver class to test the Complex class
public class Main {
    public static void main(String[] args) {
        Complex c1 = new Complex(10, 15);
        Complex c2 = new Complex(10, 15);
        if (c1 == c2) {
            System.out.println("Equal ");
        } else {
            System.out.println("Not Equal ");
        }
    }
}
```

Output:

Not Equal

The reason for printing “Not Equal” is simple: when we compare c1 and c2, it is checked whether both c1 and c2 refer to same object or not (object variables are always references in Java). c1 and c2 refer to two different objects, hence the value (c1 == c2) is false. If we create another reference say c3 like following, then (c1 == c3) will give true.

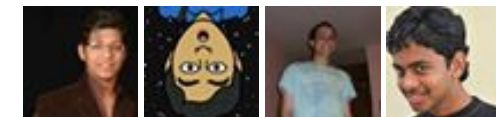
Google™ Custom Search



GeeksforGeeks



53,527 people like GeeksforGeeks.



Facebook

Interview Experiences

Advanced Data Structures

Dynamic Programming

Greedy Algorithms

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

Geometric Algorithms

```
Complex c3 = c1; // (c3 == c1) will be true
```

So, how do we check for equality of values inside the objects? All classes in Java inherit from the Object class, directly or indirectly (See point 1 of [this](#)). The **Object class** has some basic methods like clone(), toString(), equals(),.. etc. We can override the equals method in our class to check whether two objects have same data or not.

```
class Complex {  
    private double re, im;  
  
    public Complex(double re, double im) {  
        this.re = re;  
        this.im = im;  
    }  
  
    // Overriding equals() to compare two Complex objects  
    @Override  
    public boolean equals(Object o) {  
  
        // If the object is compared with itself then return true  
        if (o == this) {  
            return true;  
        }  
  
        /* Check if o is an instance of Complex or not  
        "null instanceof [type]" also returns false */  
        if (!(o instanceof Complex)) {  
            return false;  
        }  
  
        // typecast o to Complex so that we can compare data members  
        Complex c = (Complex) o;  
  
        // Compare the data members and return accordingly  
        return Double.compare(re, c.re) == 0  
            && Double.compare(im, c.im) == 0;  
    }  
}  
  
// Driver class to test the Complex class  
public class Main {  
  
    public static void main(String[] args) {  
        Complex c1 = new Complex(10, 15);
```



## Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

```

Complex c2 = new Complex(10, 15);
if (c1.equals(c2)) {
    System.out.println("Equal ");
} else {
    System.out.println("Not Equal ");
}
}
}

```

Output:

Equal

As a side note, when we override equals(), it is recommended to also override the hashCode() method. If we don't do so, equal objects may get different hash-values; and hash based collections, including HashMap, HashSet, and Hashtable do not work properly (see [this](#) for more details). We will be covering more about hashCode() in a separate post.

References:

[Effective Java Second Edition](#)

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**HIGH-PERFORMANCE COMPUTING ON A UNIVERSITY BUDGET**

Define your ideal server

**Download**

**SERVERSDIRECT**

# Deploy Early. Deploy Often.

DevOps from  
Rackspace:

## Automation

**FIND OUT HOW ▶**

**rackspace**  
the open cloud company



## Related Topics:

- Inner class in java
- How to compare two arrays in Java?
- Can we override private methods in Java?
- Can we Overload or Override static methods in java ?
- Static class in Java
- Do we need forward declarations in Java?
- Checked vs Unchecked Exceptions in Java
- Bitwise right shift operators in Java



2



Tweet

0



0

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

8 Comments

GeeksforGeeks

Sort by Newest ▾



Join the discussion...



**YourJavaTutor** · 2 years ago

Thanks.Please visit <http://yourjavatutor.blogspot....> for more explanation.

^ | ▾ · Reply · Share ›



**Radhakrishnan C H** · 2 years ago

Thanks.Please visit <http://yourjavatutor.blogspot....> for more explanation.

^ | ▾ · Reply · Share ›

## Recent Comments

Abhi You live US or India?

[Google \(Mountain View\) interview](#) · 38 minutes ago

**Aman** Hi, Why arent we checking for conditions...

[Write a C program to Delete a Tree.](#) · 1 hour ago

kzs please provide solution for the problem...

[Backtracking | Set 2 \(Rat in a Maze\)](#) · 1 hour ago

**Sanjay Agarwal** bool

tree::Root\_to\_leaf\_path\_given\_sum(tree...

[Root to leaf path sum equal to a given number](#) · 1

hour ago

**GOPI GOPINATH** @admin Highlight this sentence "We can easily...

[Count trailing zeroes in factorial of a number](#) · 1

hour ago

**newCoder3006** If the array contains negative numbers also. We...

[Find subarray with given sum](#) · 2 hours ago

AdChoices ▶

▶ [Java Array](#)

▶ [Equals Java](#)

▶ [Java Overriding](#)



**rahsgu** · 2 years ago

insted of writing new method it can be over ridden this has best examples on \

^ | v · Reply · Share ›



**shadabmast** · 2 years ago

@kartik - perfect as per Effective Java :-).

It may not appear obvious to many people that

instanceof return false for null object, so it will be helpful for them if you could r  
this :

```
[sourcecode language="java"]
/* Check if o is an instance of Complex or not.
"null instanceof [type]" also returns false
*/
```

^ | v · Reply · Share ›



**sekhar** · 2 years ago

```
[sourcecode language="java"]
if (!(o instanceof Complex)) {
return false;
}
```

In the above code, if the object o is an instance of a sub class of Complex clas  
is it still returns true..

^ | v · Reply · Share ›



**kartik** → sekhar · 2 years ago

It will not go inside this if condition. It will compare members (re and im  
will return true. If a derived class wants to compare derived class spec  
the derived class to override the equals(). Inside the equals() of deriver  
super.equals() and then derived class specific data.

^ | v · Reply · Share ›

AdChoices ▶

▶ [Java Override](#)

▶ [Java Object](#)

▶ [String Java](#)

AdChoices ▶

▶ [Java Args](#)

▶ [Mit Java](#)

▶ [Java Null](#)



krishna · 2 years ago

good info!

Instead of overriding equals(), a new method can also be created and new me equal. Is there any problem with this new method approach?

^ | v · Reply · Share ›



kartik → krishna · 2 years ago

Instead of overriding equals(), we can write our own methods. In gener as this is the method generally used for comparing data, your class us equals. Please correct me if I am wrong.

1 ^ | v · Reply · Share ›



Subscribe



Add Disqus to your site

@geeksforgeeks, **Some rights reserved**

**Contact Us!**

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team