

* It is used to compare two Objects based on **memory address OR memory reference**

It is a predefined non static method of Object class.

* It is used to compare two Objects based on **memory address OR memory reference**

* Internally, It uses == operator to compare the address/reference of two Objects.

```
//Programs
package com.ravi.equals_demo;

class Customer
{
    private int id;
    private String name;

    public Customer(int id, String name)
    {
        super();
        this.id = id;
        this.name = name;
    }
}

public class EqualsDemo1
{
    public static void main(String[] args)
    {
        Customer c1 = new Customer(111, "Scott");
        Customer c2 = new Customer(111, "Scott");

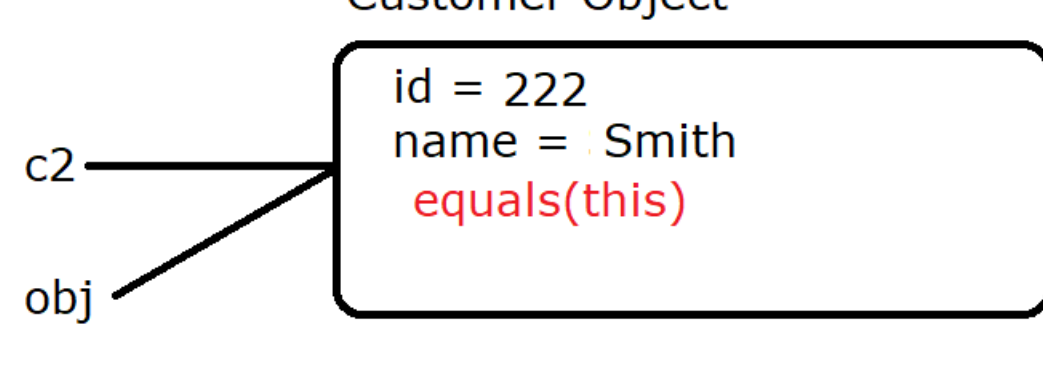
        System.out.println(c1==c2); //false

        System.out.println(c1.equals(c2)); //False
    }
}
```

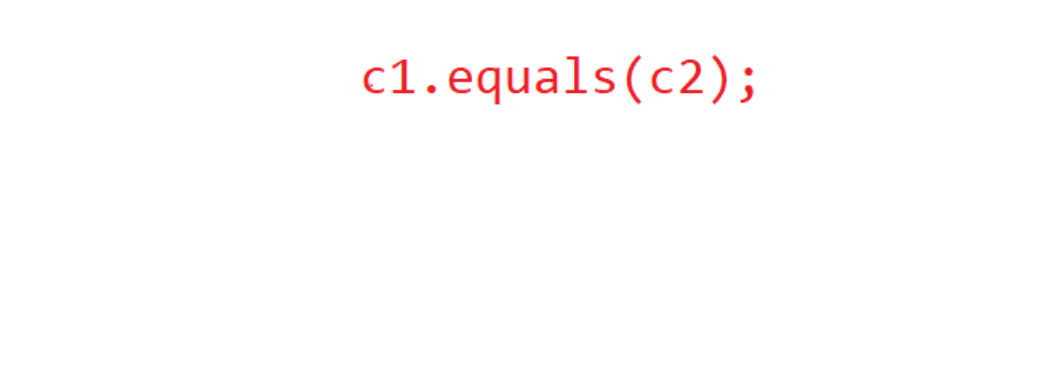
* When we call equals method of Object class then we are getting the output as false even both the objects are content wise same so IF WE WANT TO COMPARE BOTH THE OBJECTS **BASED ON THE CONTENT** (NOT BASED ON THE MEMORY ADDRESS) then we should override equals(Object obj) method from Object class.

How to compare two Objects based on the content ?

Customer c1 = new Customer(111, "Scott");



Customer c2 = new Customer(222, "Smith");



```
//Overriding equals() method for Content Compariosn

@Override
public boolean equals(final Customer this, Object obj)
{
    //Retrieve both the objects data for content comparison

    //First Object data
    int id1 = this.id;
    String name1 = this.name;

    //Second Object data
    Customer c2 = (Customer) obj; //Down casting
    int id2 = c2.id;
    String name2 = c2.name;

    if(id1==id2 && name1.equals(name2))
    {
        return true;
    }
    else
    {
        return false;
    }
}

System.out.println(c1.equals(c2)); //
```

c1.equals(c2);

* THERE IS A CONTRACT BETWEEN HASHCODE AND EQUALS METHOD.

* WE SHOULD ALWAYS OVERRIDE BOTH THE METHODS [NOT A SINGLE METHOD]

* WE TWO OBJECTS ARE SAME AS PER equals(Object obj) methods then Both the objects must produce same hash code.

```
package com.ravi.equals_demo;

class Customer
{
    private int id;
    private String name;

    public Customer(int id, String name)
    {
        super();
        this.id = id;
        this.name = name;
    }

    @Override
    public int hashCode()
    {
        return this.id;
    }

    //Overriding for Content Comparison
    @Override
    public boolean equals(Object obj) //obj = c2
    {
        //retrieveing the data of both the objects for Comparison

        //First Object data
        int id1 = this.id;
        String name1 = this.name;

        //Second Object data
        Customer c2 = (Customer) obj; //Down casting
        int id2 = c2.id;
        String name2 = c2.name;

        if(id1==id2 && name1.equals(name2))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

public class EqualsDemo1
{
    public static void main(String[] args)
    {
        Customer c1 = new Customer(111, "Scott");
        Customer c2 = new Customer(111, "Scott");

        System.out.println(c1.equals(c2)); //true

        System.out.println(c1.hashCode()+" : "+c2.hashCode());
    }
}
```

```
package com.ravi.equals_demo;

class Product
{
    private int id;
    private String name;

    public Product(int id, String name)
    {
        super();
        this.id = id;
        this.name = name;
    }

    //I want to compare both the objects based on the Content
    //So, I need to override both the methods i.e hashCode() and equals()

    @Override
    public int hashCode()
    {
        return this.id;
    }

    @Override
    public boolean equals(Object obj) //obj = p2
    {
        Product p2 = (Product) obj;

        if(this.id == p2.id && this.name.equals(p2.name))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

public class EqualsDemo2 {

    public static void main(String[] args)
    {
        Product p1 = new Product(111, "Camera");
        Product p2 = new Product(111, "Camera");

        System.out.println(p1.equals(p2));
        System.out.println(p1.hashCode() +" : "+p2.hashCode() );

    }

}
```

String class and all the Wrapper classes have overridden hashCode() and equals(Object obj) method from Object class.

```
package com.ravi.equals_demo;

import java.util.Objects;

class Student
{
    private int studentId;
    private String studentName;

    public Student(int studentId, String studentName)
    {
        super();
        this.studentId = studentId;
        this.studentName = studentName;
    }

    @Override
    public int hashCode()
    {
        return this.studentId;
    }

    @Override
    public boolean equals(Object obj) //obj = employee obj
    {
        if(obj instanceof Student)
        {
            Student s2 = (Student) obj;

            if(this.studentId==s2.studentId && this.studentName.equals(s2.studentName))
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        else
        {
            System.err.println("Comparison is not possible");
            return false;
        }
    }
}

public class EqualsDemo3
{
    public static void main(String[] args)
    {
        Student s1 = new Student(111,"Raj");
        Employee e1 = new Employee(111, "Raj");

        System.out.println(s1.equals(e1));
        Student s2 = new Student(111, "Raj");
        System.out.println(".....");
        System.out.println(s1.equals(s2));
    }
}

class Employee
{
    private int employeeId;
    private String employeeName;

    public Employee(int employeeId, String employeeName)
    {
        super();
        this.employeeId = employeeId;
        this.employeeName = employeeName;
    }
}
```