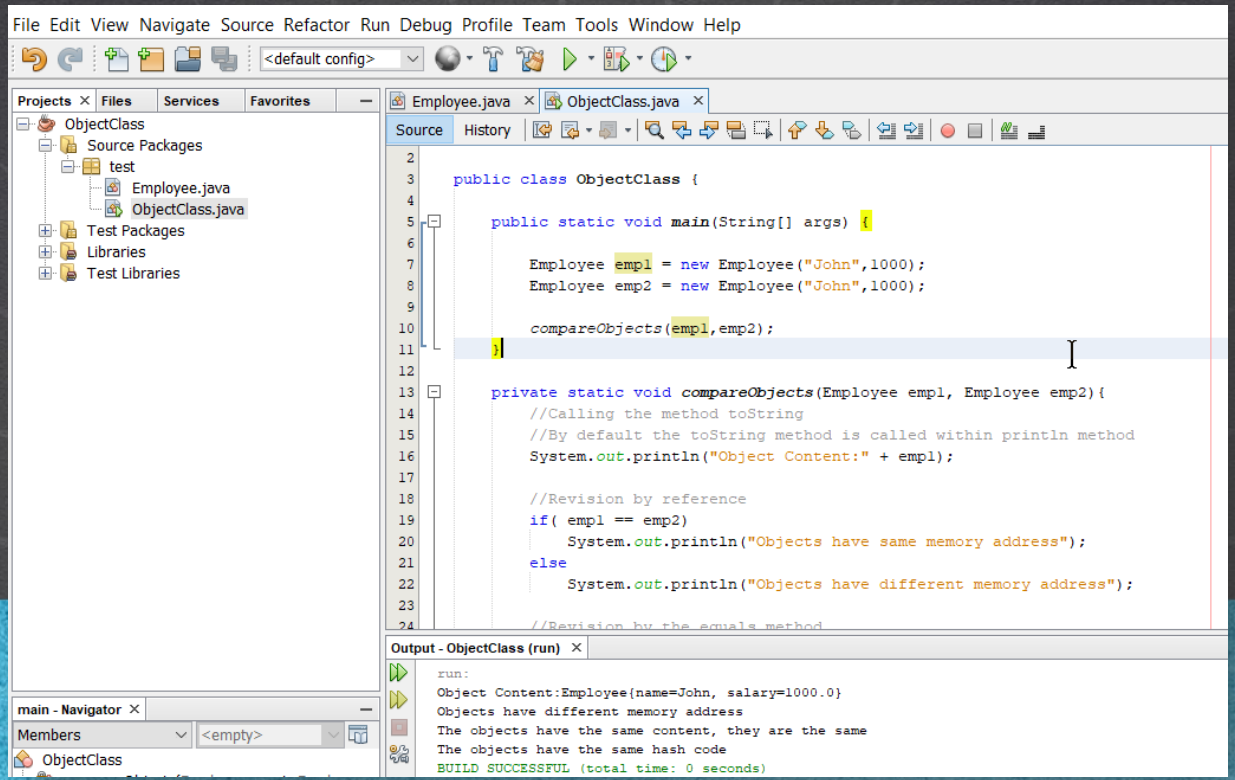
JAVA PROGRAMMING COURSE

EXERCISE

OBJECT CLASS: TOSTRING, EQUALS, AND HASCODE METHODS

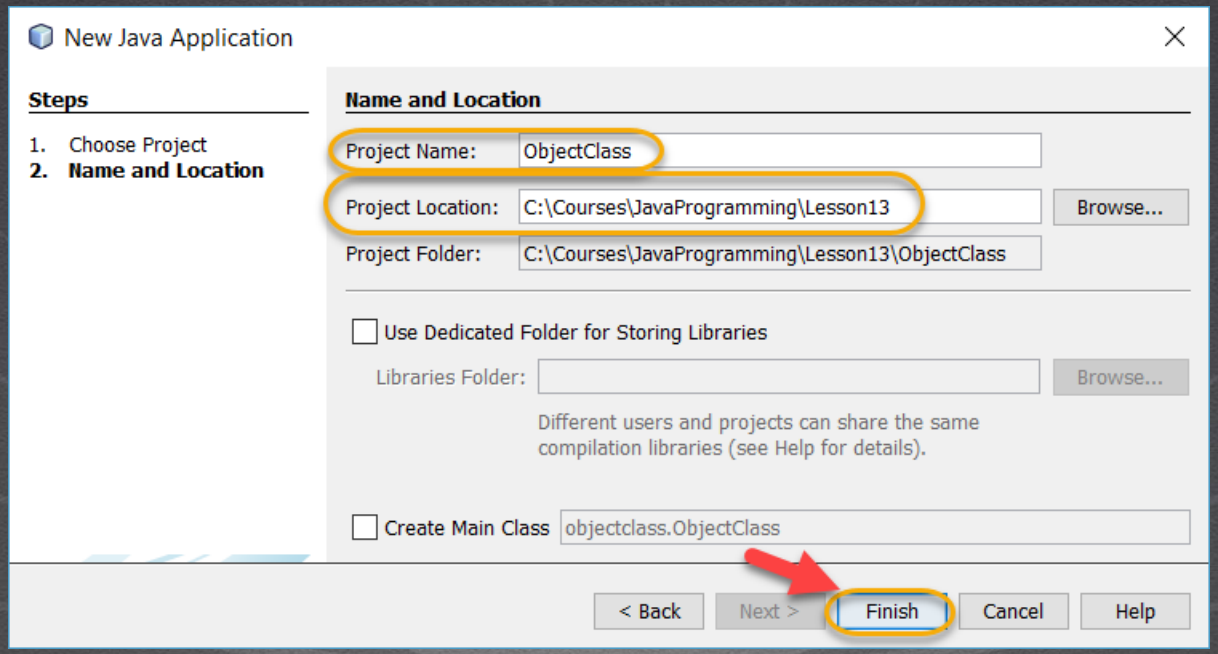
EXERCISE OBJECTIVE

Create the exercise of the Object Class. At the end we should observe the following :



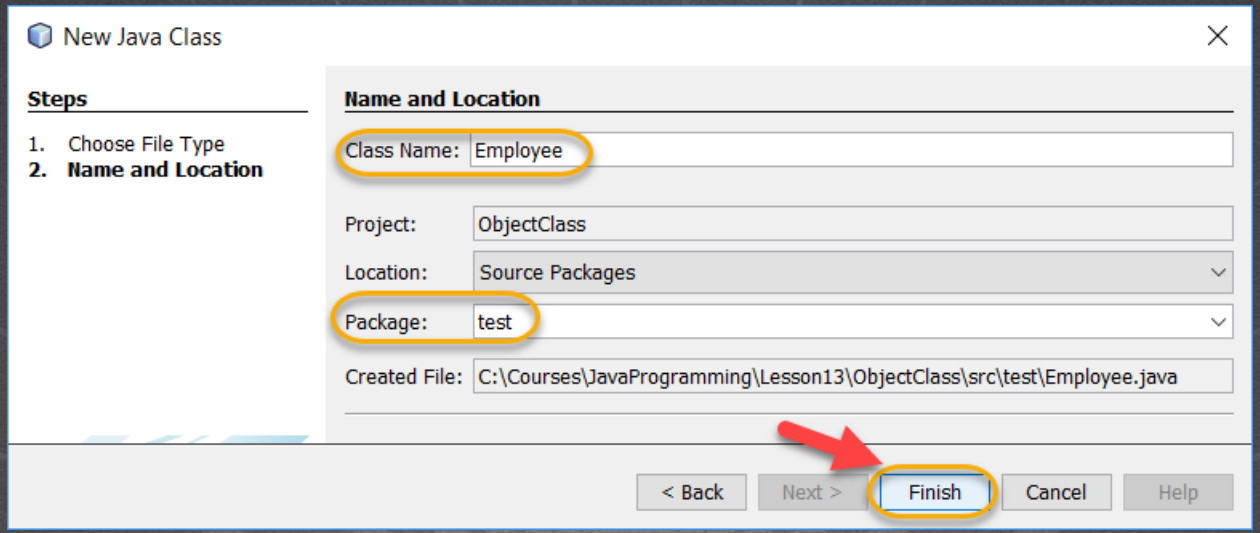
1. CREATE A NEW PROJECT

Create a new project:



2. CREATE A NEW CLASS

Create a new class:



3. MODIFY THE CODE

Employee.java:

```
package test;

public class Employee {

    private String name;
    private double salary;

    Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getSalary() {
        return salary;
    }
```

3. MODIFY THE CODE

Employee.java:

```
public void setSalary(double salary) {
    this.salary = salary;
}

@Override
public String toString() {
    return "Employee{" + "name=" + name + ", salary=" + salary + '}';
}

@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (obj instanceof Employee) {
        Employee emp = (Employee) obj;
        if (name.equals(emp.name) && Double.valueOf(salary).equals(emp.salary)) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

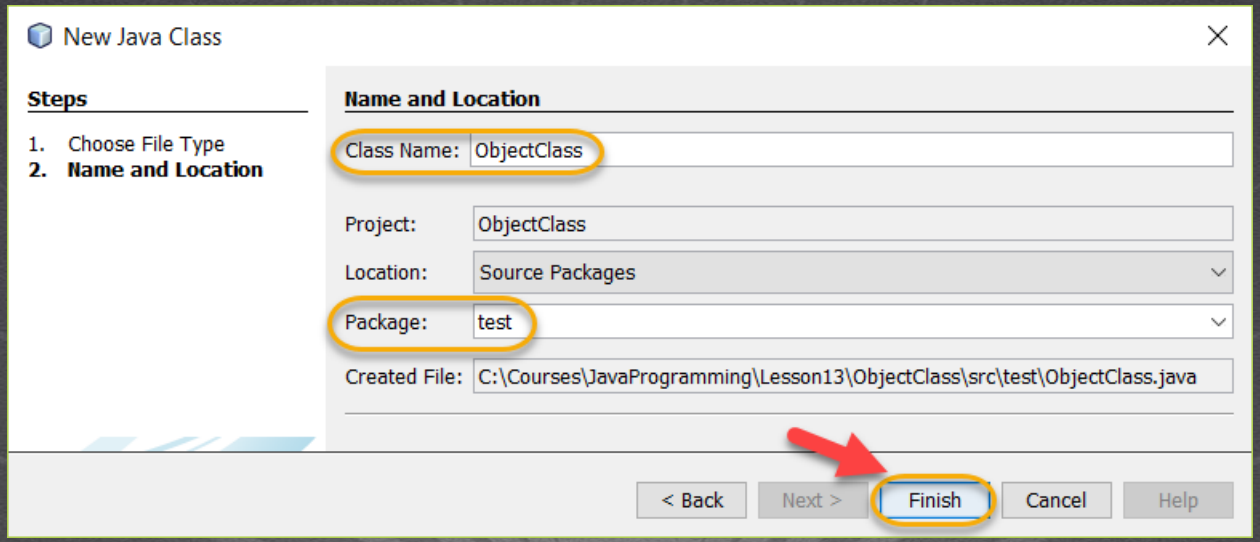
3. MODIFY THE CODE

Employee.java:

```
    @Override
    public int hashCode() {
        int hash = 7;
        hash = 31 * hash + this.name.hashCode();
        hash = 31 * hash + Double.valueOf(this.salary).hashCode();
        return hash;
    }
}
```

4. CREATE A NEW CLASS

Create a new class:



5. MODIFY THE CODE

ObjectClass.java:

```
package test;

public class ObjectClass {

    public static void main(String[] args) {

        Employee emp1 = new Employee("John",1000);
        Employee emp2 = new Employee("John",1000);

        compareObjects(emp1,emp2);
    }

    private static void compareObjects(Employee emp1, Employee emp2){
        //Calling the method toString
        //By default the toString method is called within println method
        System.out.println("Object Content:" + emp1);

        //Revision by reference
        if( emp1 == emp2)
            System.out.println("Objects have same memory address");
        else
            System.out.println("Objects have different memory address");
```

5. MODIFY THE CODE

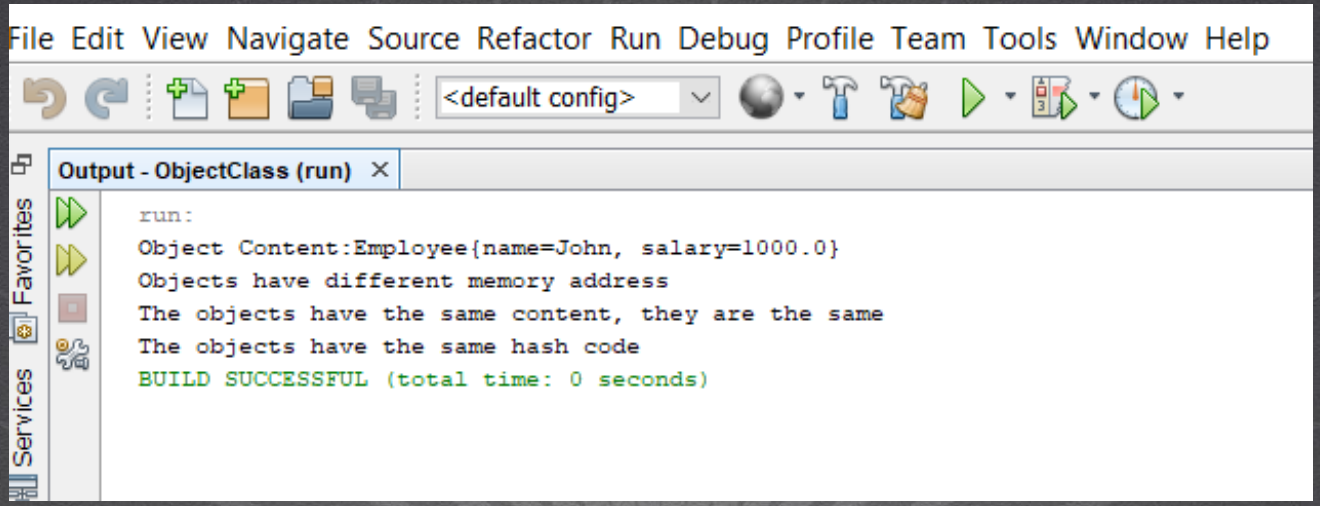
ObjectClass.java:

```
        //Revision by the equals method
        if(emp1.equals(emp2))
            System.out.println("The objects have the same content, they are the same");
        else
            System.out.println("The objects do NOT have the same content, they are NOT the same");

        //We review the hashCode method
        if(emp1.hashCode() == emp2.hashCode())
            System.out.println("The objects have the same hash code");
        else
            System.out.println("The objects do NOT have the same hash code");
    }
}
```

6. EXECUTE THE PROJECT

The result is as follows:



EXERCISE CONCLUSION

- With this exercise we have put into practice the handling of the Object class, as well as the overriding of some of the most important methods, such as the toString, equals and hashCode methods.

- For more information consult:
- https://docs.oracle.com/javase/tutorial/java/IandI/objectclass.html

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta