1. No, the **calculateTax()** method is not a pure function.

   A pure function must satisfy the following conditions:

   - Always return the same output for the same input.
   - Have no side effects and not depend on external or mutable state.

   Here **rate** is an instance variable, so the result of calculateTax(amount) depends on the state of the object, not just on the input amount. This means if the **rate** is changed in another part of the program, the method could return different results for the same amount.

   To make it a pure function we need to refactor the method so that it uses only parameters, not instance variables:

   Modified code

   ```
   public class TaxUtil {
      public double calculateTax(double amount, double rate) {
         return amount * rate;
      }
   }
   ```

   Github link :
   https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task1/src

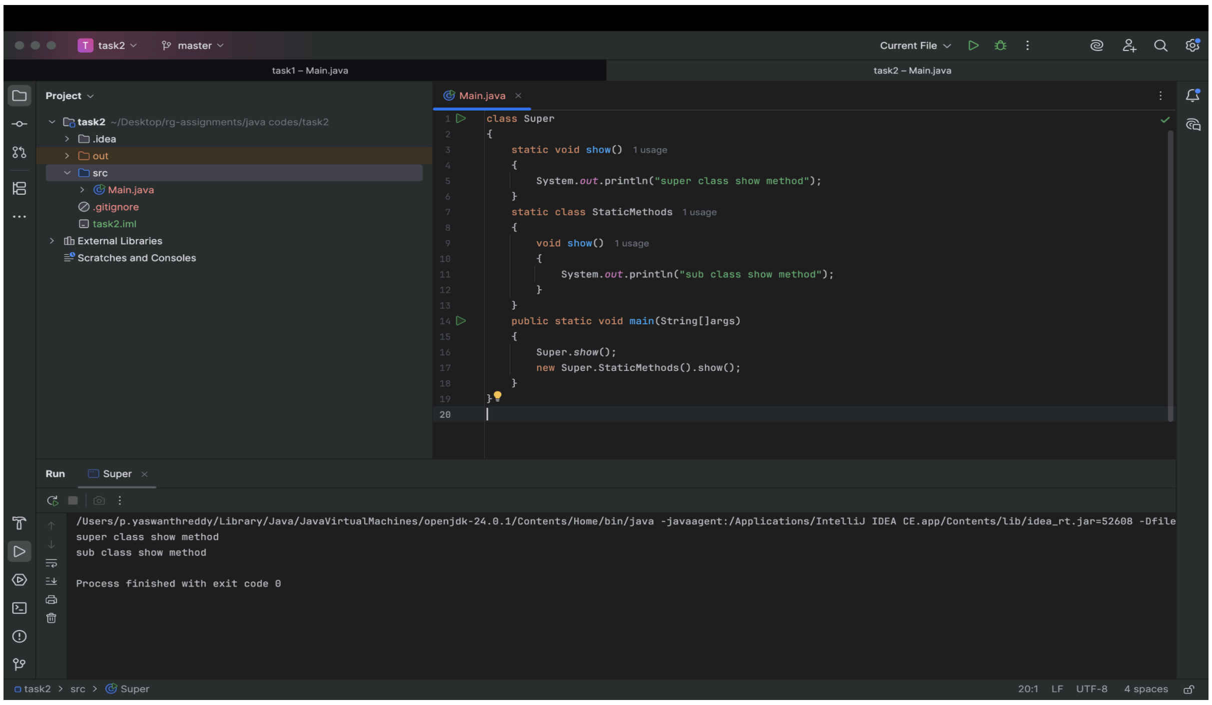
2. The output for the given code is attached in the screenshot below

   The **Super.show()** method calls the static method **show()** of the Super class.
   So the output would be : super class show method

   The **new Super.StaticMethods().show()** method instantiates the static inner class **StaticMethods** and calls its non-static method **show().**
   So the output would be : sub class show method.

   Github link:
   https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task2/src

Output

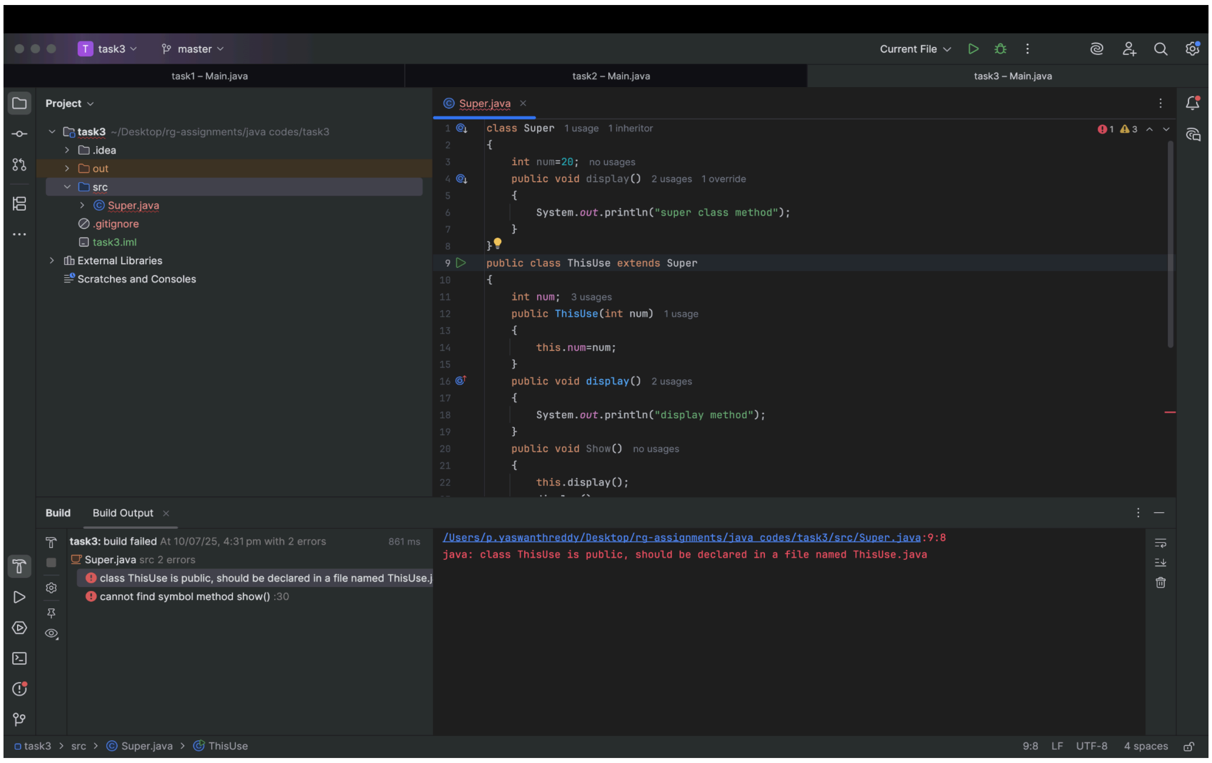


```java
class Super
{
    static void show()
    {
        System.out.println("super class show method");
    }
    static class StaticMethods
    {
        void show()
        {
            System.out.println("sub class show method");
        }
    }
    public static void main(String[]args)
    {
        Super.show();
        new Super.StaticMethods().show();
    }
}
```

```
/Users/p.yaswanthreddy/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52608 -Dfile
super class show method
sub class show method

Process finished with exit code 0
```

3. The output for the given code is as follows

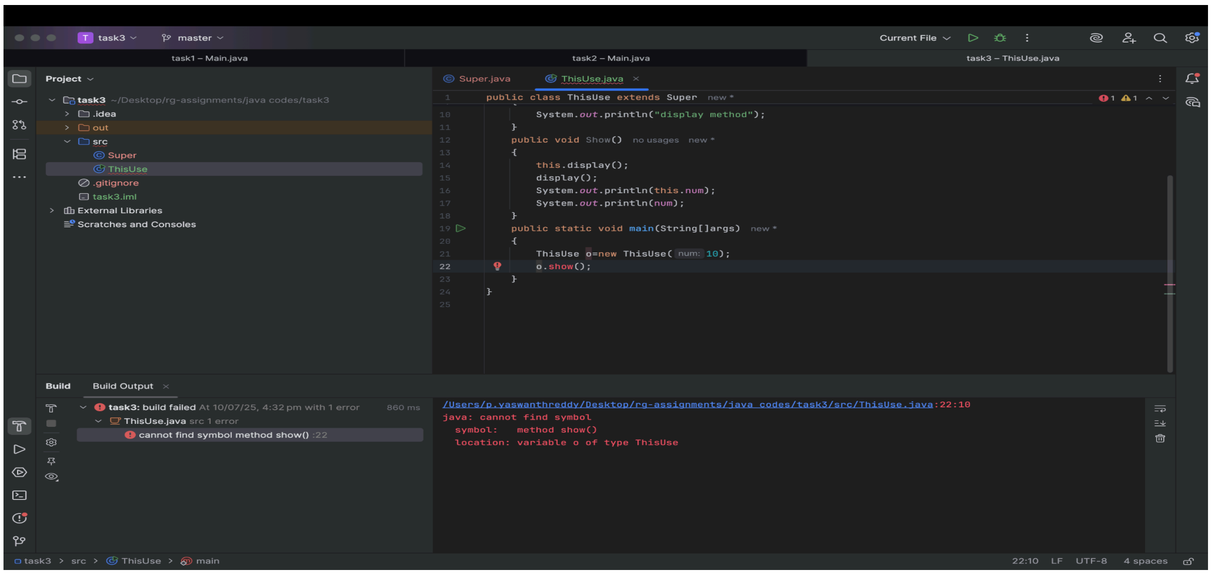
Case1: If both **Super** class and **ThisUse** class are defined in the same file we get the following error “class ThisUse is public, should be declared in a file named ThisUse.java”



```java
class Super
{
    int num=20;
    public void display()
    {
        System.out.println("super class method");
    }
}
public class ThisUse extends Super
{
    int num;
    public ThisUse(int num)
    {
        this.num=num;
    }
    public void display()
    {
        System.out.println("display method");
    }
    public void Show()
    {
        this.display();
```

```
task3: build failed At 10/07/25, 4:31 pm with 2 errors    861 ms
Super.java src 2 errors
  class ThisUse is public, should be declared in a file named ThisUse.j
  cannot find symbol method show() :30

/Users/p.yaswanthreddy/Desktop/rg-assignments/java codes/task3/src/Super.java:9:8
java: class ThisUse is public, should be declared in a file named ThisUse.java
```

Case2: If both the **Super** and **ThisUse** classes are defined in different files we get the following error " cannot find symbol method show(), location : variable o of type ThisUse"



```java
public class ThisUse extends Super
        System.out.println("display method");
    }
    public void Show()
    {
        this.display();
        display();
        System.out.println(this.num);
        System.out.println(num);
    }
    public static void main(String[]args)
    {
        ThisUse o=new ThisUse( num: 10);
        o.show();
    }
}
```

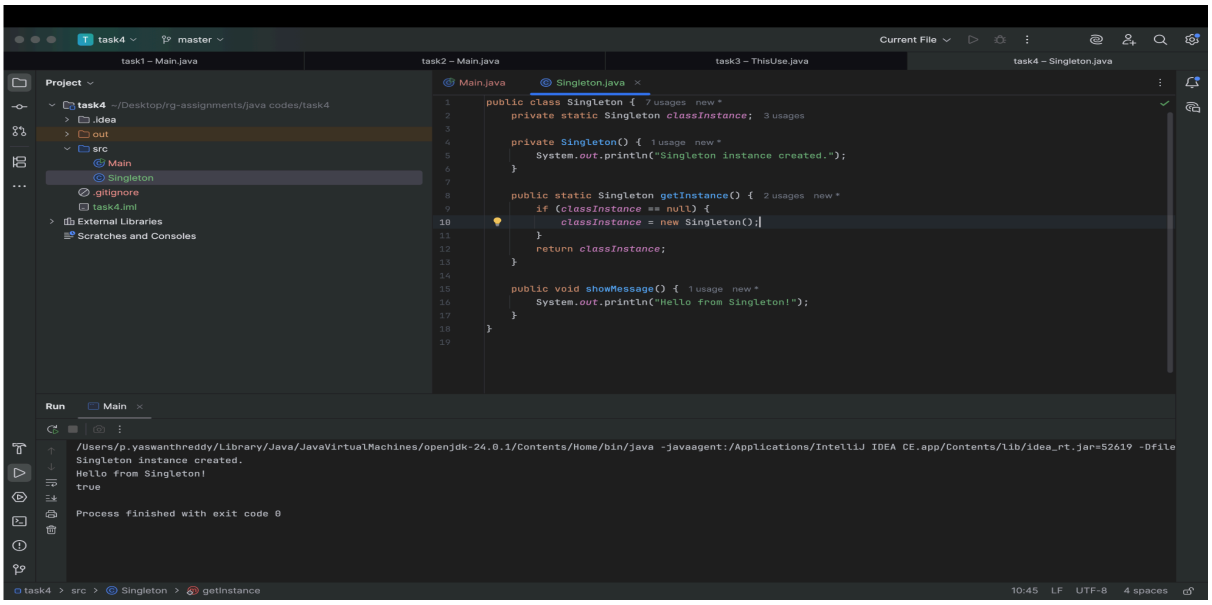
```
task3: build failed At 10/07/25, 4:32 pm with 1 error   860 ms
  ThisUse.java src 1 error
    cannot find symbol method show() :22
```

```
/Users/p.yaswanthreddy/Desktop/rg-assignments/java codes/task3/src/ThisUse.java:22:10
java: cannot find symbol
  symbol:   method show()
  location: variable o of type ThisUse
```

Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task3/src

4. A Singleton design pattern is a design pattern that ensures a class has only one instance and provides a global access point to it. It's often used when you need to control the instantiation of a class and ensure that only one object of that type exists throughout the application's lifecycle. It can be used to control access to shared resources like databases, logging and configuration.

Code & output screenshot



```java
public class Singleton {
    private static Singleton classInstance;

    private Singleton() {
        System.out.println("Singleton instance created.");
    }

    public static Singleton getInstance() {
        if (classInstance == null) {
            classInstance = new Singleton();
        }
        return classInstance;
    }

    public void showMessage() {
        System.out.println("Hello from Singleton!");
    }
}
```

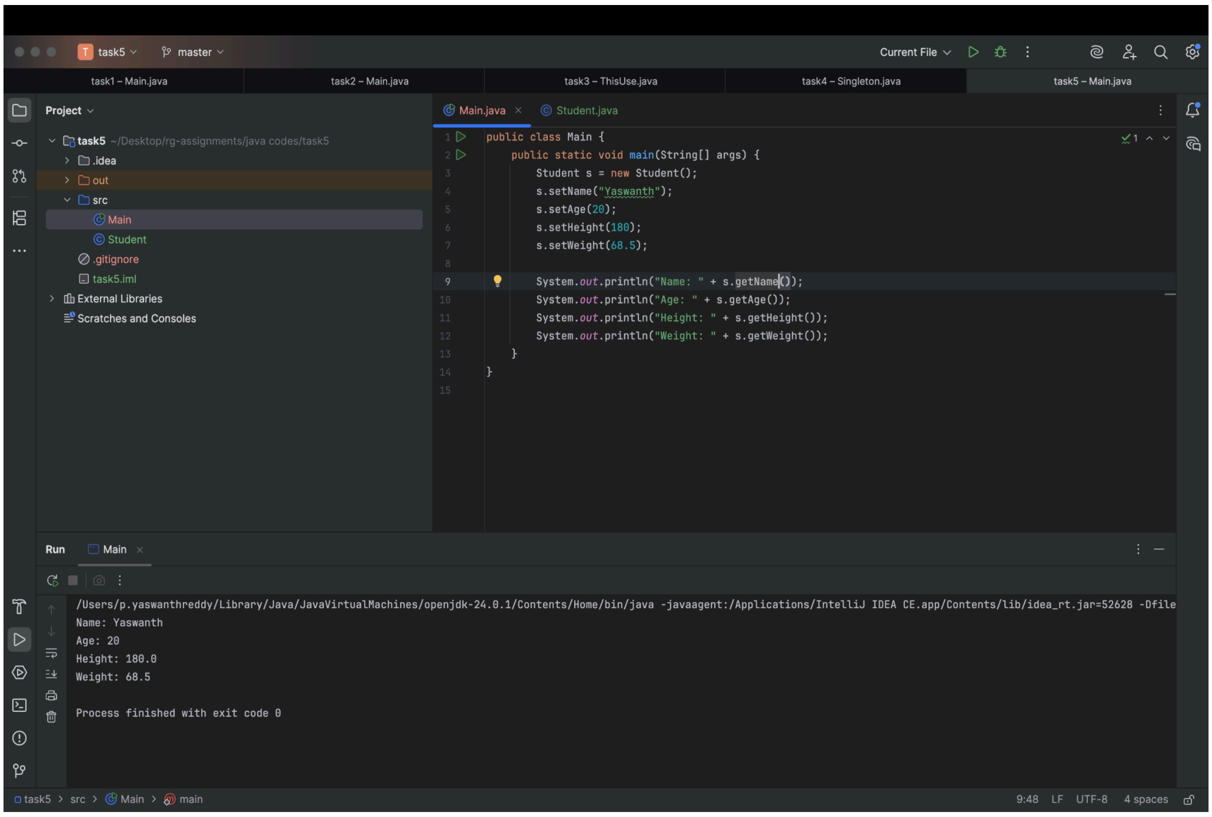
```
/Users/p.yaswanthreddy/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52619 -Dfile
Singleton instance created.
Hello from Singleton!
true

Process finished with exit code 0
```

Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task4/src

5. Encapsulation is the process in which we bind the data members and methods into a single unit. It is used to hide internal data of a class and expose only what's necessary via methods.

   It can be achieved by the following way:
   - Make instance variables private.
   - Provide public getter and setter methods to access/update the instance variable.

   Code and output screenshot

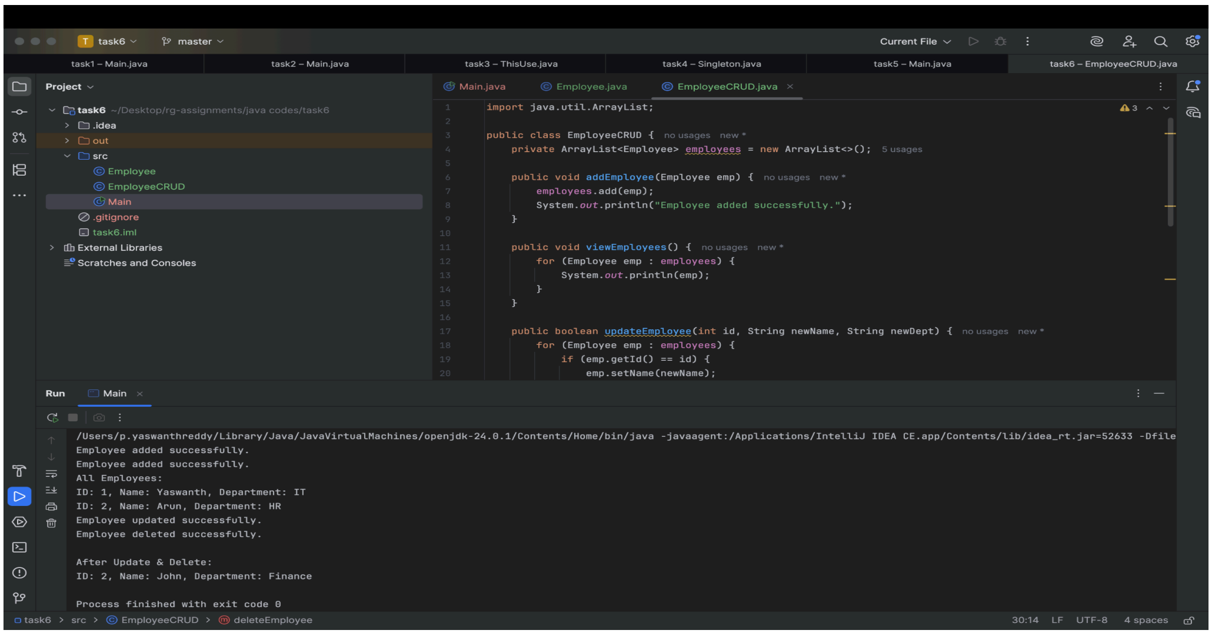


```java
public class Main {
    public static void main(String[] args) {
        Student s = new Student();
        s.setName("Yaswanth");
        s.setAge(20);
        s.setHeight(180);
        s.setWeight(68.5);

        System.out.println("Name: " + s.getName());
        System.out.println("Age: " + s.getAge());
        System.out.println("Height: " + s.getHeight());
        System.out.println("Weight: " + s.getWeight());
    }
}
```

```
/Users/p.yaswanthreddy/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52628 -Dfile
Name: Yaswanth
Age: 20
Height: 180.0
Weight: 68.5

Process finished with exit code 0
```

   Github link:
   https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task5/src

6. Code & output screenshot

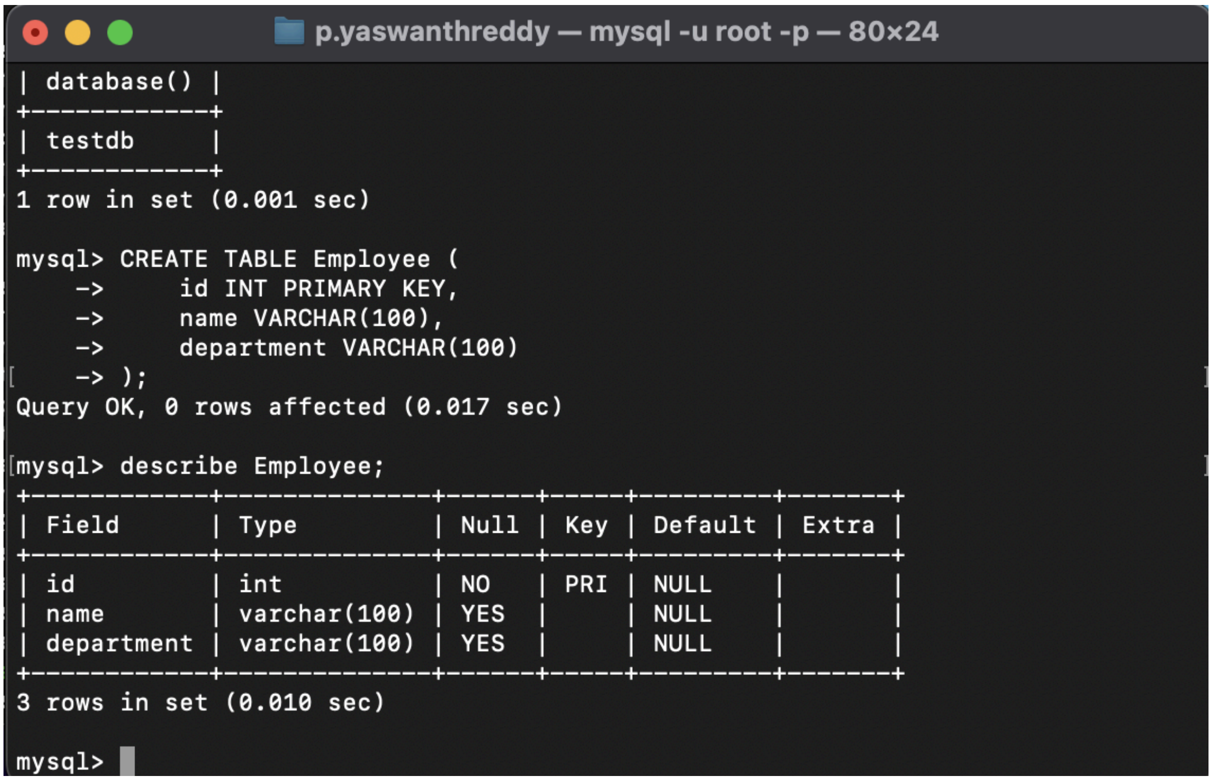


```
import java.util.ArrayList;

public class EmployeeCRUD {
    private ArrayList<Employee> employees = new ArrayList<>();

    public void addEmployee(Employee emp) {
        employees.add(emp);
        System.out.println("Employee added successfully.");
    }

    public void viewEmployees() {
        for (Employee emp : employees) {
            System.out.println(emp);
        }
    }

    public boolean updateEmployee(int id, String newName, String newDept) {
        for (Employee emp : employees) {
            if (emp.getId() == id) {
                emp.setName(newName);
```

```
/Users/p.yaswanthreddy/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52633 -Dfile
Employee added successfully.
Employee added successfully.
All Employees:
ID: 1, Name: Yaswanth, Department: IT
ID: 2, Name: Arun, Department: HR
Employee updated successfully.
Employee deleted successfully.

After Update & Delete:
ID: 2, Name: John, Department: Finance

Process finished with exit code 0
```

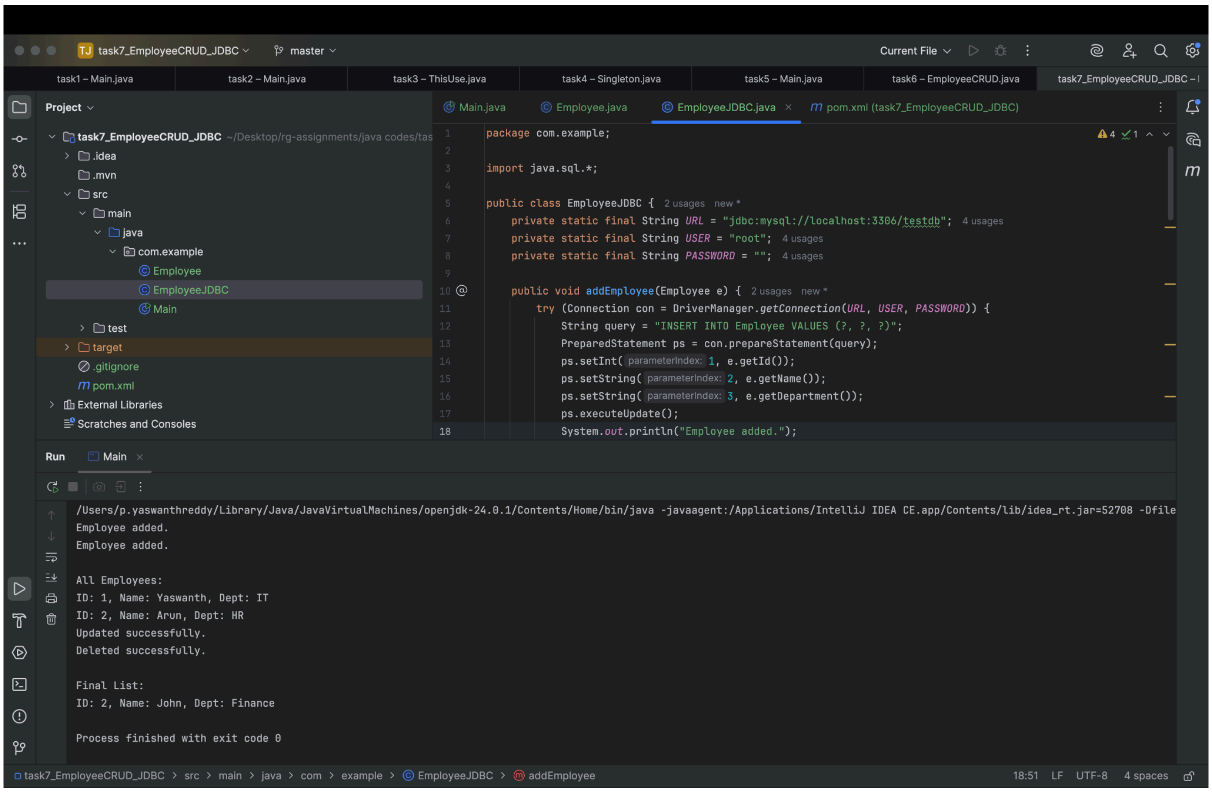
Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task6/src

7. Employee table creation



```
| database() |
+------------+
| testdb     |
+------------+
1 row in set (0.001 sec)

mysql> CREATE TABLE Employee (
    ->     id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     department VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.017 sec)

mysql> describe Employee;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| id         | int          | NO   | PRI | NULL    |       |
| name       | varchar(100) | YES  |     | NULL    |       |
| department | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
3 rows in set (0.010 sec)

mysql> 
```

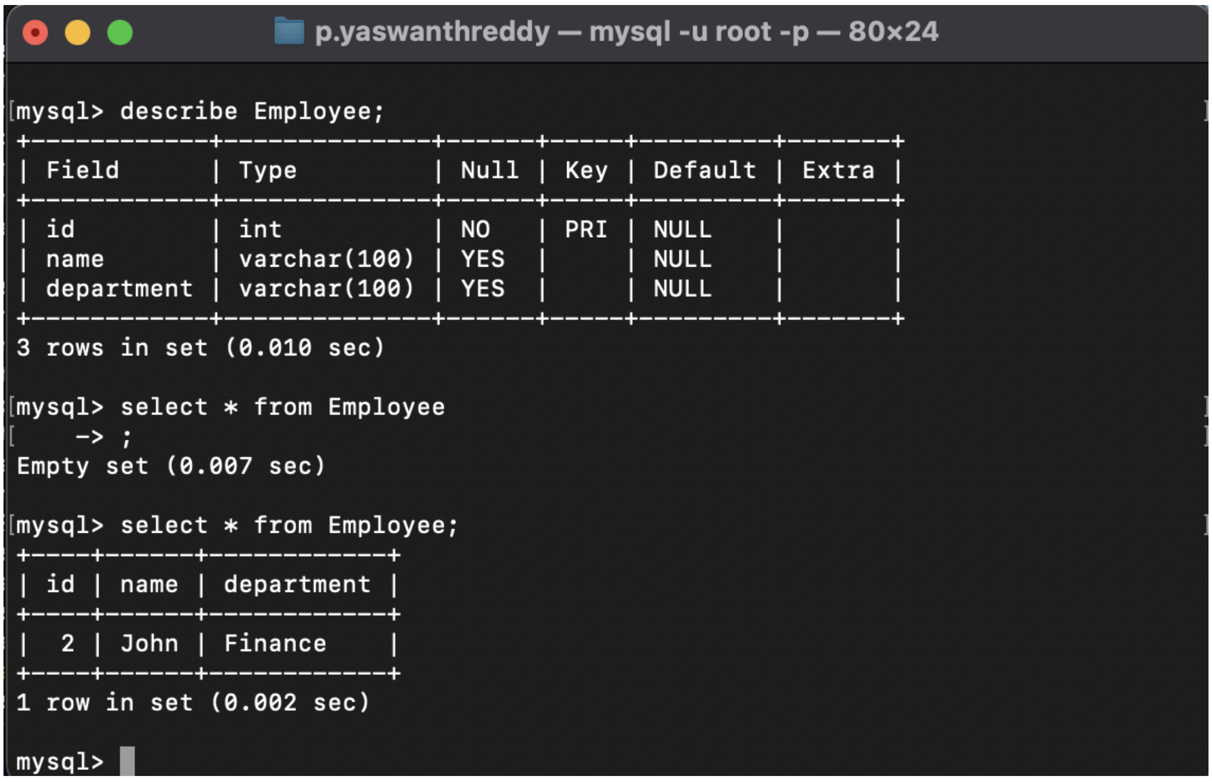
Code and output screenshot



```java
package com.example;

import java.sql.*;

public class EmployeeJDBC {
    private static final String URL = "jdbc:mysql://localhost:3306/testdb";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public void addEmployee(Employee e) {
        try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
            String query = "INSERT INTO Employee VALUES (?, ?, ?)";
            PreparedStatement ps = con.prepareStatement(query);
            ps.setInt( parameterIndex: 1, e.getId());
            ps.setString( parameterIndex: 2, e.getName());
            ps.setString( parameterIndex: 3, e.getDepartment());
            ps.executeUpdate();
            System.out.println("Employee added.");
```

```
/Users/p.yaswanthreddy/Library/Java/JavaVirtualMachines/openjdk-24.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=52708 -Dfile
Employee added.
Employee added.

All Employees:
ID: 1, Name: Yaswanth, Dept: IT
ID: 2, Name: Arun, Dept: HR
Updated successfully.
Deleted successfully.

Final List:
ID: 2, Name: John, Dept: Finance

Process finished with exit code 0
```

Data in Employee table after CRUD operations



```
mysql> describe Employee;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| id         | int          | NO   | PRI | NULL    |       |
| name       | varchar(100) | YES  |     | NULL    |       |
| department | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
3 rows in set (0.010 sec)

mysql> select * from Employee
    -> ;
Empty set (0.007 sec)

mysql> select * from Employee;
+----+------+------------+
| id | name | department |
+----+------+------------+
|  2 | John | Finance    |
+----+------+------------+
1 row in set (0.002 sec)

mysql>
```

Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/tree/master/java%20codes/task7_EmployeeCRUD_JDBC/src/main/java/com/example