Please make sure to save/push all your code in the branch feature-java created in the previous week assignment as part of your github repo rg-assignments

Please share your output screenshots in the assignment document along with the github link for each question. Provide an explanation wherever possible as part of your response :-)

1)

Given:

```
public class TaxUtil {
  double rate = 0.15;

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

Would you consider the method calculateTax() a 'pure function'? Why or why not?If you claim the method is NOT a pure function, please suggest a way to make it pure.

Sol:
The method calculateTac() is not considered as a Pure function because it depends on the external variable rate to calculate the tax amount.
In order to convert the method as a Pure function then we have to pass the rate variable as a parameter to the calculateTax() method.
Updated code:

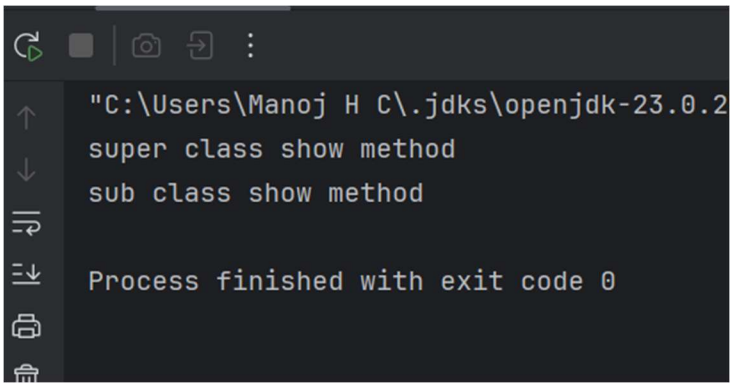
```
public class TaxUtil {

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

2)
What will be the output for following code?

```
class Super
{
static void show()
{
System.out.println("super class show method");
}
static class StaticMethods
{
```

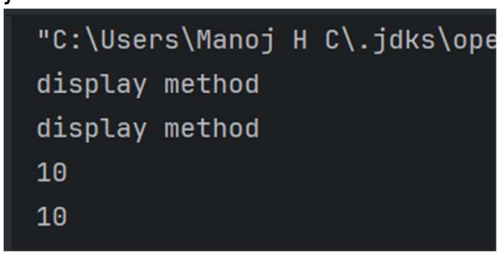
```
void show()
{
System.out.println("sub class show method");
}
}
public static void main(String[]args)
{
Super.show();
new Super.StaticMethods().show();
}
}
```



3)
What will be the output for the following code?

```
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();
display();
System.out.println(this.num);
System.out.println(num);
}
public static void main(String[]args)
{
ThisUse o=new ThisUse(10);
o.show();
}
}
```

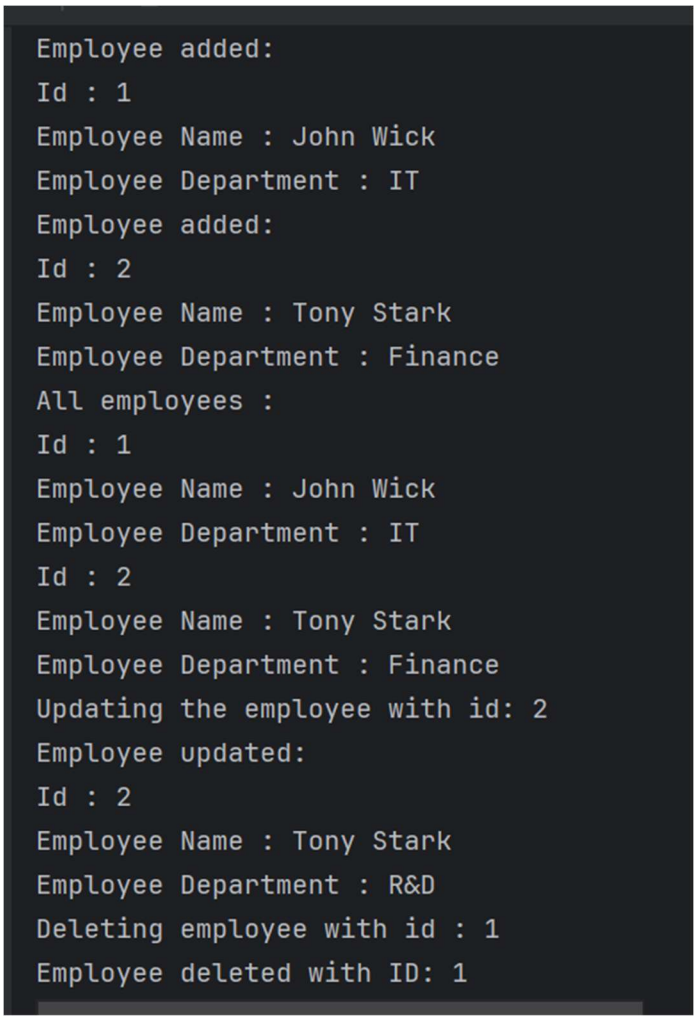


4) What is the singleton design pattern? Explain with a coding example.
Sol:
Singleton Pattern is a design pattern which is used to make sure that only one instance of a class is created and used throughout the application. some of the examples are the database connection code.

5) How do we make sure a class is encapsulated? Explain with a coding example.
Sol:
A class can be encapsulated by marking the variable fields of that class as private and creating a getter and setter methods to operate on that data. The getters and setters have a public visibility

6)
Perform CRUD operation using ArrayList collection in an EmployeeCRUD class for the below Employee

```
class Employee{
        private int id;
        private String name;
        private String department;
}
```



7) Perform CRUD operation using JDBC in an EmployeeJDBC class for the below Employee

```
class Employee{
        private int id;
        private String name;
        private String department;
}
```

```
"C:\Users\Manoj H C\.jdks\openjdk-23.0.2\bin\java.exe" "-javaagent:
Employee added:
Id : 5
Employee Name : Captain America
Employee Department : Marvel
Employee added:
Id : 6
Employee Name : Bruce Wayne
Employee Department : DC
All employees :
Employee{id=2, name='Bruce Wayne', department='DC'}
Employee{id=3, name='Captain America', department='Avengers'}
Employee{id=5, name='Captain America', department='Marvel'}
Employee{id=6, name='Bruce Wayne', department='DC'}
Updating employee :
Employee updated.
Deleting Employee
Employee deleted.

Process finished with exit code 0
```