Core Java Assignment

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.

Ans. TaxUtil() is not a pure function. A pure function should always return the same output for the same input, and should not depend on or modify any state outside of its scope.

Corrected code->

```
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();
}
}
```

Ans. super class show method

sub class show method

----

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();
}
}
```

Ans. display method

display method

10

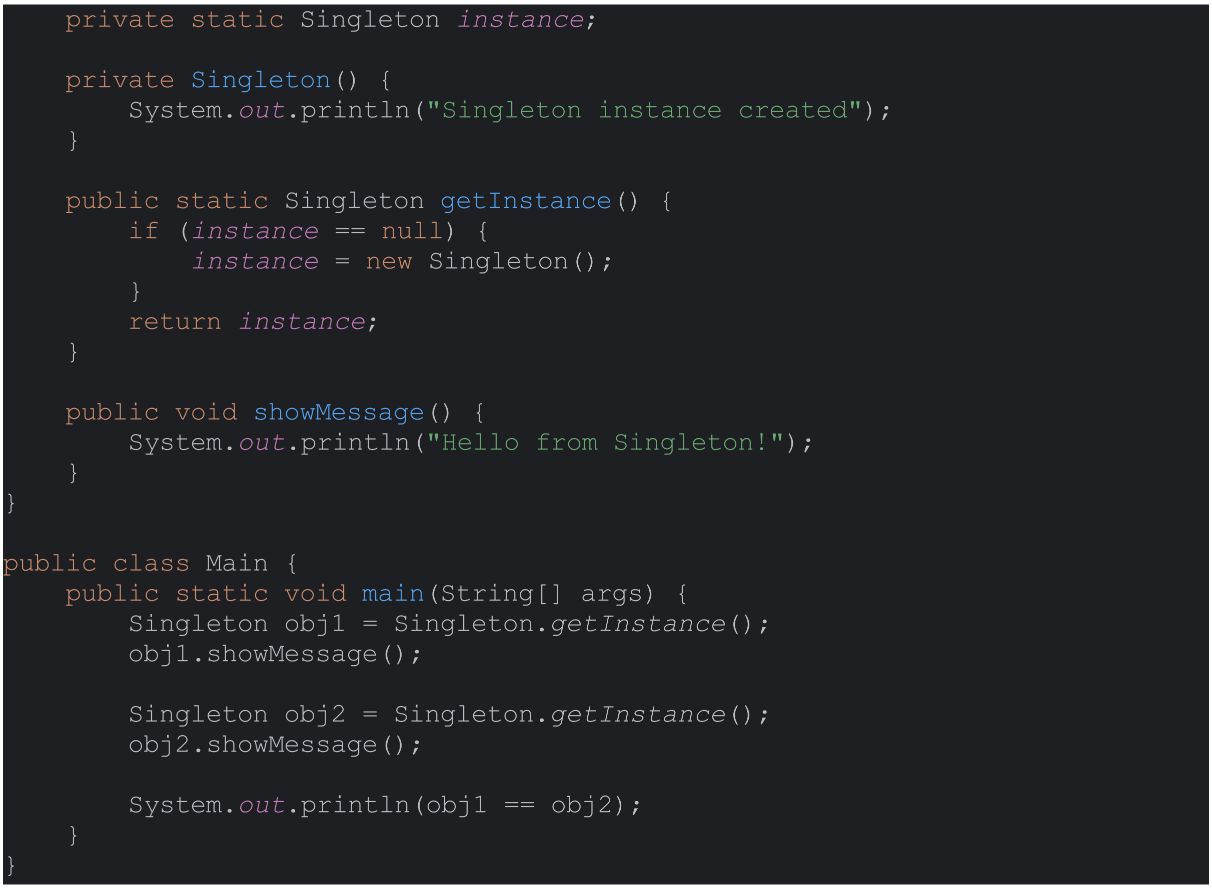
10

---

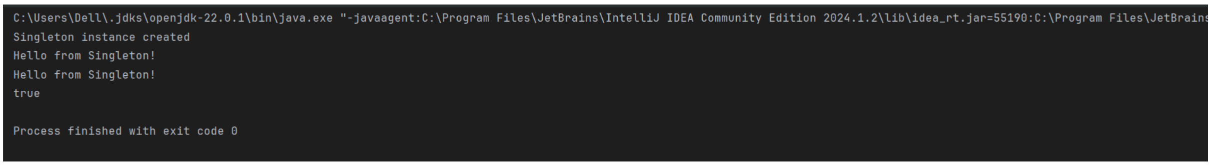
4) What is the singleton design pattern? Explain with a coding example.

Ans. Singleton ensures that only one instance of a class is created and provides a global point of access to it.

```
class Singleton {
```



```
    private static Singleton instance;

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

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

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

public class Main {
    public static void main(String[] args) {
        Singleton obj1 = Singleton.getInstance();
        obj1.showMessage();

        Singleton obj2 = Singleton.getInstance();
        obj2.showMessage();

        System.out.println(obj1 == obj2);
    }
}
```

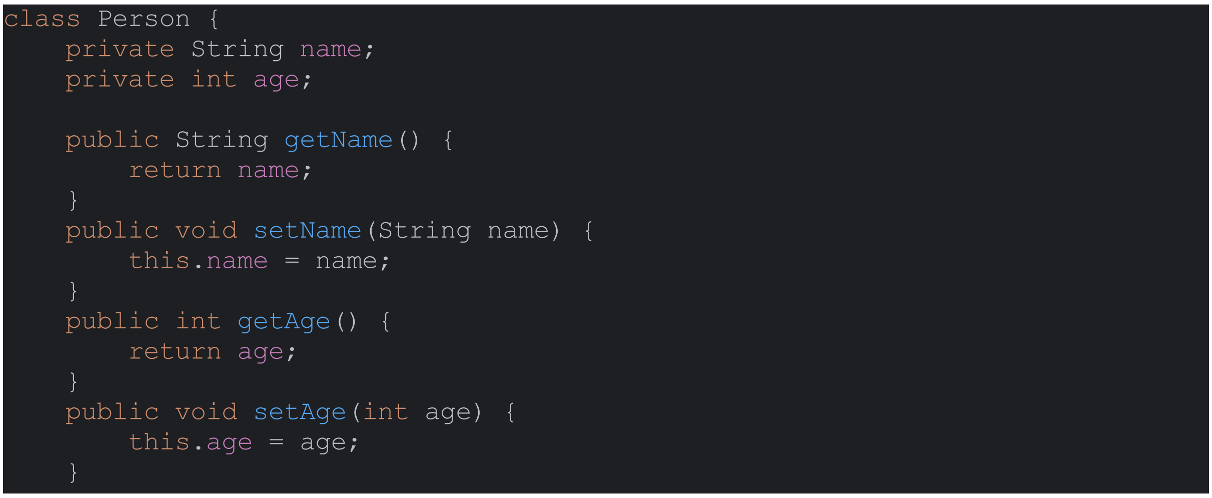


```
C:\Users\Dell\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\lib\idea_rt.jar=55190:C:\Program Files\JetBrains
Singleton instance created
Hello from Singleton!
Hello from Singleton!
true

Process finished with exit code 0
```

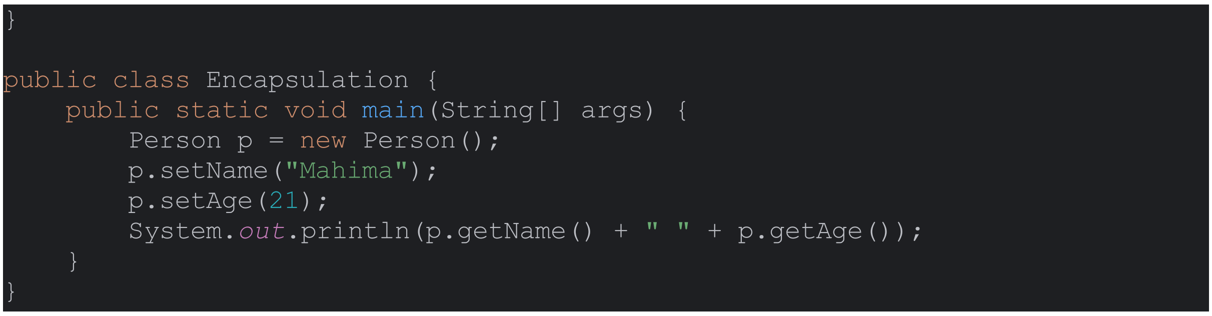
---

5) How do we make sure a class is encapsulated? Explain with a coding example.

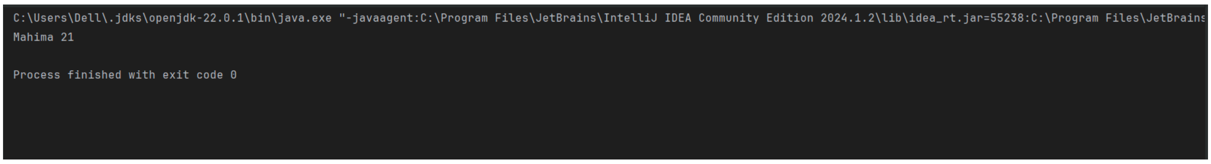
Ans. Encapsulation means hiding internal details and providing access via public methods with the help of getters and setters.



```
class Person {
    private String name;
    private int age;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
```



```
}

public class Encapsulation {
    public static void main(String[] args) {
        Person p = new Person();
        p.setName("Mahima");
        p.setAge(21);
        System.out.println(p.getName() + " " + p.getAge());
    }
}
```

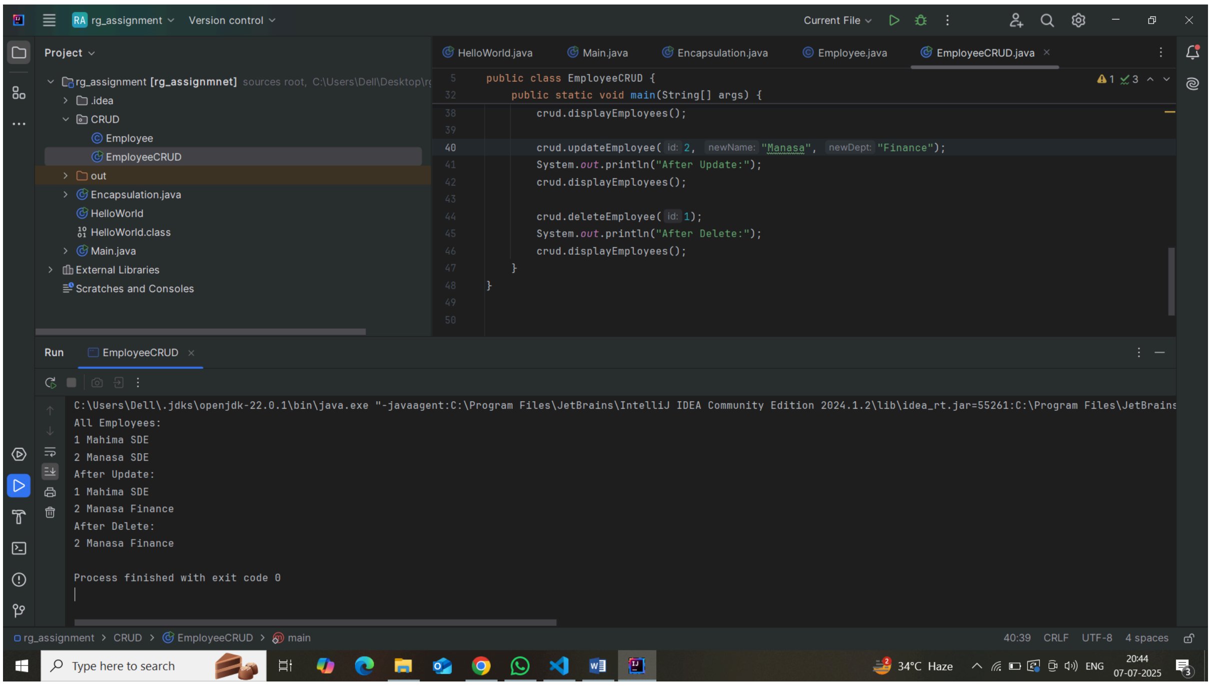


```
C:\Users\Dell\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\lib\idea_rt.jar=55238:C:\Program Files\JetBrains
Mahima 21

Process finished with exit code 0
```

----

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;
}
```



```
public class EmployeeCRUD {
    public static void main(String[] args) {
        crud.displayEmployees();

        crud.updateEmployee(2, "Manasa", "Finance");
        System.out.println("After Update:");
        crud.displayEmployees();

        crud.deleteEmployee(1);
        System.out.println("After Delete:");
        crud.displayEmployees();
    }
}
```

```
C:\Users\Dell\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\lib\idea_rt.jar=55261:C:\Program Files\JetBrains
All Employees:
1 Mahima SDE
2 Manasa SDE
After Update:
1 Mahima SDE
2 Manasa Finance
After Delete:
2 Manasa Finance

Process finished with exit code 0
```

Ans. https://github.com/mahima23anu/rg_assignments/tree/feature-java/CRUD

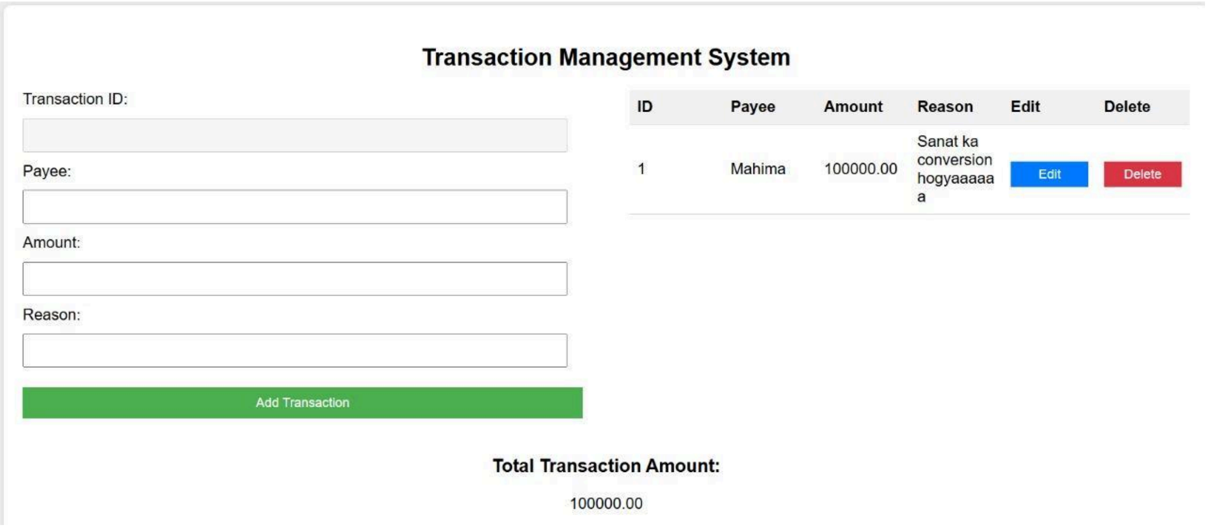
----

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;
}
```

Ans.
https://github.com/mahima23anu/rg_assignments/tree/feature-java/CRUD_WITH_JDBC

Javascript Assignment



Transaction Management System

Transaction ID:

Payee:

Amount:

Reason:

ID	Payee	Amount	Reason	Edit	Delete
1	Mahima	100000.00	Sanat ka conversion hogyaaaaa a	Edit	Delete

Total Transaction Amount:

100000.00

Github link : https://github.com/mahima23anu/rg_assignments/tree/feature-java/