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.

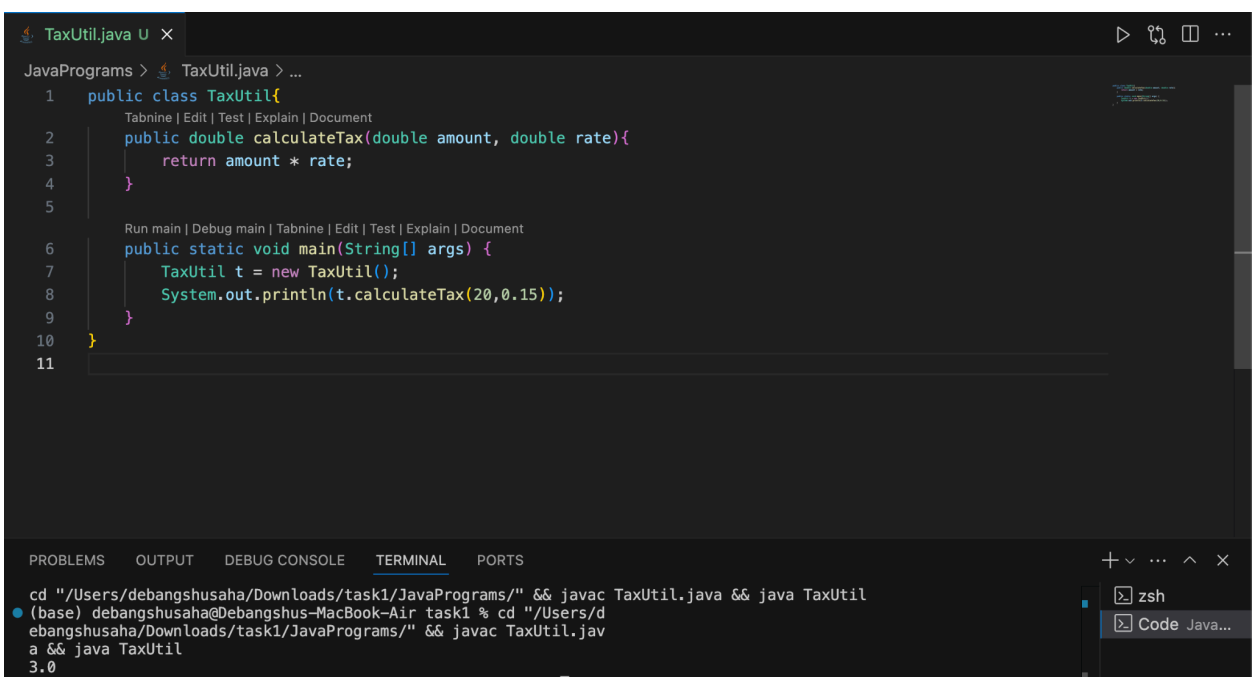
-> No, the method calculateTax() in its current form **is not a pure function**. Because,

- A **pure function** should rely **only on its input parameters** and **should not depend on or modify any external state**.
- calculateTax() depends on the **instance variable rate**, which is external to the method's parameters.
    - If someone changes rate, the output of calculateTax() will change, **even though the input (amount) stays the same.**
    - This violates the principle of referential transparency.

We can accept the rate as a parameter along with the amount, to make it a pure function

Github - https://github.com/DebangshuSaha2002/rg-assignments/blob/feature-java/Assignment2/TaxUtil.java

Screenshot:

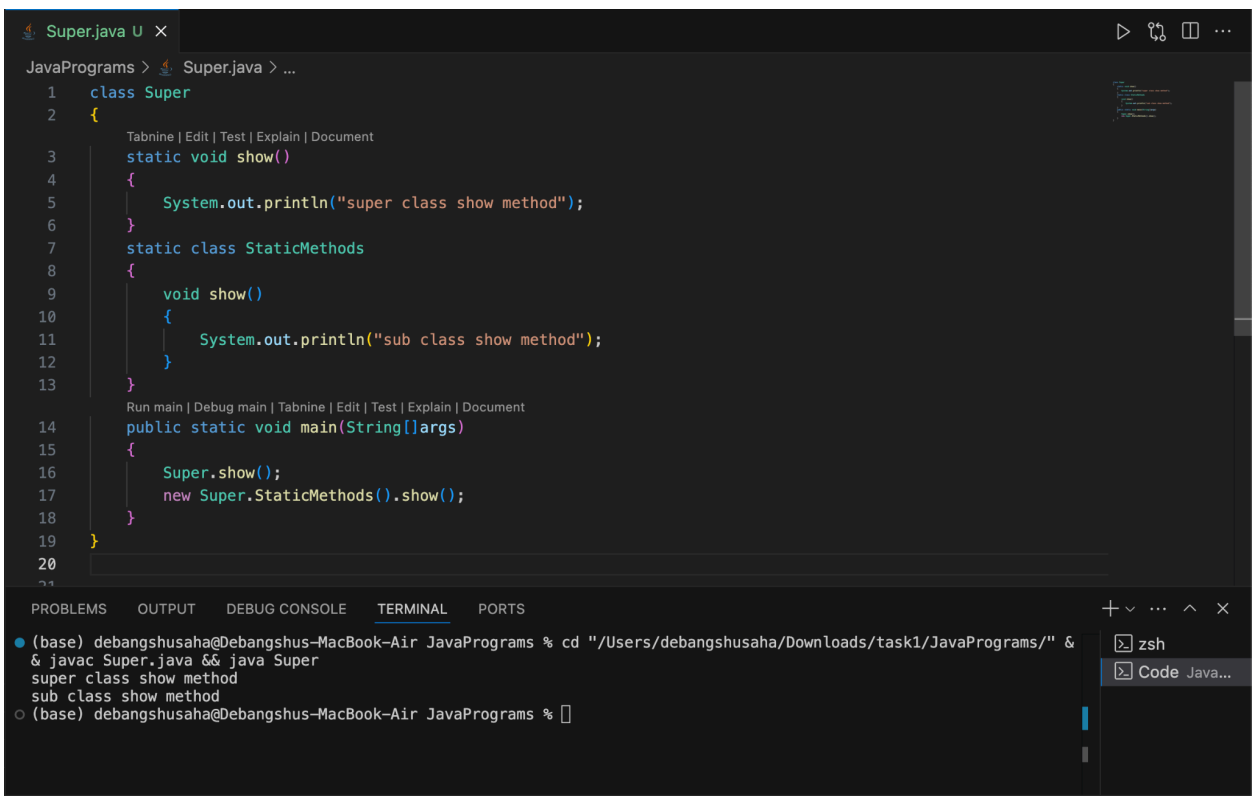


2) What will be the output for the following code?

```
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();
        }
}
```

Github : https://github.com/DebangshuSaha2002/rg-assignments/blob/feature-java/Assignment2/Super.java

Screenshot:

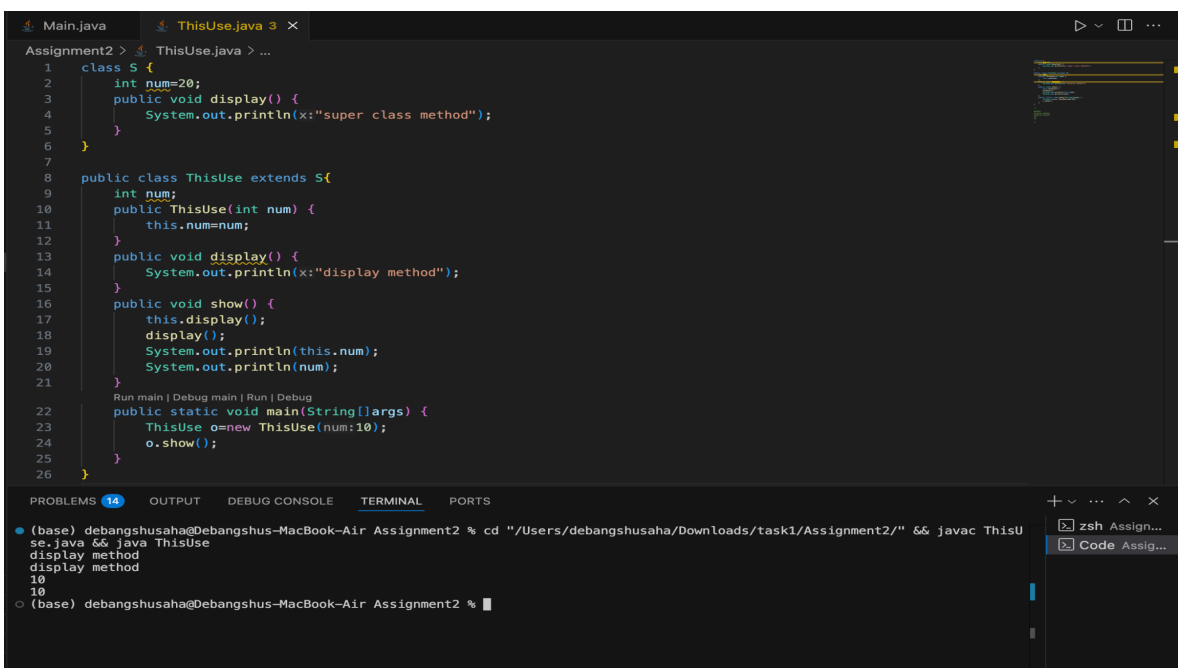


3) What will be the output for the following code?

```
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();
	}
}
```

-> Github : https://github.com/DebangshuSaha2002/rg-assignments/blob/feature-java/Assignment2/ThisUse.java

Screenshot:



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

-> Singleton Design Pattern is one of the simplest design patterns. It ensures that a class is having only one instance of it, and it provides a global point of access to it.
To implement a singleton design pattern, we need to make sure that we use a private constructor.

Singleton design pattern code - https://github.com/DebangshuSaha2002/rg-assignments/blob/feature-java/Assignment2/singleTonDesignPattern/DP.java

DP.java 1 × threadSafeDP.java 4

JavaPrograms > singleTonDesignPattern > DP.java > DP

```java
//This method of doing the singleton pattern works fine but it is not thread safe
//So to make it thread safe we can use the double checked locking method

class A{
    public static A obj;

    //Declare the constructor as private
    private A(){
        System.out.println("Inside class A is constructor");
    }

    Tabnine | Edit | Test | Explain | Document
    public static A getInstance(){
        if(obj==null){
            obj=new A();
        }
        return obj;
    }
}

class DP{
    Run main | Debug main | Tabnine | Edit | Test | Explain | Document
    public static void main(String[] args) {
        A obj1 = A.getInstance();
        A obj2 = A.getInstance();

        //If we print the address to which these objects point. It will be the same
        //So singleton pattern is working fine here
        System.out.println(obj1);
        System.out.println(obj2);
    }
}
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(base) debangshusaha@Debangshus-MacBook-Air JavaPrograms % cd "/Users/debangshusaha/Downloads/task1/JavaPrograms/singleTonDesignPattern/" && javac DP
.java && java DP
Inside class A is constructor
A@7ad041f3
A@7ad041f3
(base) debangshusaha@Debangshus-MacBook-Air singleTonDesignPattern %
```

zsh
Code singl...

-> Singleton Design Pattern using the double checked locking to make it thread safe as well - https://github.com/DebangshuSaha2002/rg-assignments/blob/feature-java/Assignment2/singleTonDesignPattern/threadSafeDP.java

threadSafeDP.java 4

threadSafeDP.java > ...

```java
//Thread safe approach using the double checked locking method
class A{
    public static A obj;
    //Declare the constructor as private
    private A(){
        System.out.println("Inside class A's constructor");
    }
    Tabnine | Edit | Test | Fix | Explain | Document
    public static A getInstance(){ //Double checked locking
        if(obj==null){
            synchronized (A.class) {
                if(obj==null){
                    System.out.println("Obj is null, creating new instance by " + Thread.currentThread().getName());
                    obj = new A();
                }
            }
        }
        return obj;
    }
}
class threadSafeDP{
    Run main | Debug main | Tabnine | Edit | Test | Fix | Explain | Document
    public static void main(String[] args) {
        Thread t1 = new Thread(() -> {
            A obj1 = A.getInstance();
        }, "Thread1");
        Thread t2 = new Thread(() -> {
            A obj2 = A.getInstance();
        }, "Thread2");
        t1.start();
        t2.start();
    }
}
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(base) debangshusaha@Debangshus-MacBook-Air PracticeJava % cd "/Users/debangshusaha/Documents/Java/PracticeJava/" && javac threadSafeDP.java && java threadSafeDP
Obj is null, creating new instance by Thread1
Inside class A's constructor
(base) debangshusaha@Debangshus-MacBook-Air PracticeJava %
```

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

-> We can make sure a class is encapsulated by-

- Declaring the class variables as **private**.
- Providing **public getter** and **setter methods** to access and modify these variables.
- Keeping the business logic and data safe from unintended modification.

Github link -
https://github.com/DebangshuSaha2002/rg-assignments/blob/feature-java/Assignment2/Employee.java

Employee.java 1, U

JavaPrograms > Employee.java > Developer

```java
class Developer{

    //declaring private class variables to make sure that they are not directly accessible and hidden from outside
    private String name;
    private int age;

    //Public constructor
    public Developer(String name, int age){
        this.name = name;
        this.age = age;
    }

    //Setter for name
    Tabnine | Edit | Test | Explain | Document
    public void setName(String name){
        this.name = name;
        System.out.println("Name is set to "+name);
    }

    //Getter for name
    Tabnine | Edit | Test | Explain | Document
    public void getName(){
        System.out.println("Name: "+this.name);
    }

    //Setter for age
    Tabnine | Edit | Test | Explain | Document
    public void setAge(int age){
        //we can also add up logic in the setters
        this.age = age;
        System.out.println("Age is set to "+age);
    }

    //Getter for age
    Tabnine | Edit | Test | Explain | Document
    public void getAge(){
        System.out.println("Age: "+age);
    }
}

class Employee{
    Run main | Debug main | Tabnine | Edit | Test | Explain | Document
    public static void main(String[] args) {
        //calling the parameterized constructor to set the default values
        Developer obj = new Developer("Debangshu", 22);

        //we cannot access the class variable directly so to access the name and age we use getters
        obj.getName();
        obj.getAge();

        //and to change the class variable we use the setters
        obj.setName("Deb");
        obj.getName();

        obj.setAge(23);
        obj.getAge();
    }
}
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
  2559865..0167863  feature-java -> feature-java
(base) debangshusaha@Debangshus-MacBook-Air singleTonDesignPattern % cd "/Users/debangshusaha/Downloads/task1/JavaPrograms/" && javac Emp
loyee.java &
& java Employee
Name: Debangshu
Age: 22
Name is set to Deb
Name: Deb
```

zsh

Code Java...

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;
}
```

-> Github link -
https://github.com/DebangshuSaha2002/rg-assignments/tree/feature-java/Assignment2/CRUD_JDBC/src