**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: MCA, Semester I

**Course: Java Programming**

**Experiment No.03**

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**A.1 Aim:** To understand and implement concepts like classes, methods, objects, constructors and overloading of methods for given scenario.

**Task 1:** An object can be created of type Room and assign its address to variable rl as\_\_\_\_\_\_\_\_

1. rl = new Room();

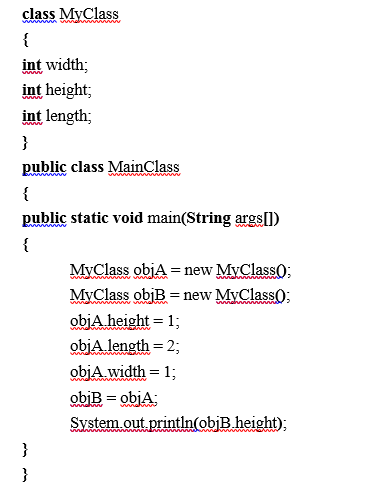
**Task 2:** Observe the following Room r2=new Room(), Variable r2 contains a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ or address of memory location where a new object is created.

c) Reference

**Task 3:** The actual \_\_\_\_\_\_\_\_\_\_\_\_\_ is contained inside the individual objects and not in the class.

1. Data

**Task 4:** **What will be the output of the following Java Code?**

****

**TASK 5: Design a class to represent a student. Include the following data members**

**Name of the student, Roll no of the student and Total Marks.**

**Include methods to assign values and display values for the student.**

**Task 6: Design a class to represent a bank account. Include the following members:**

**Data Member:**

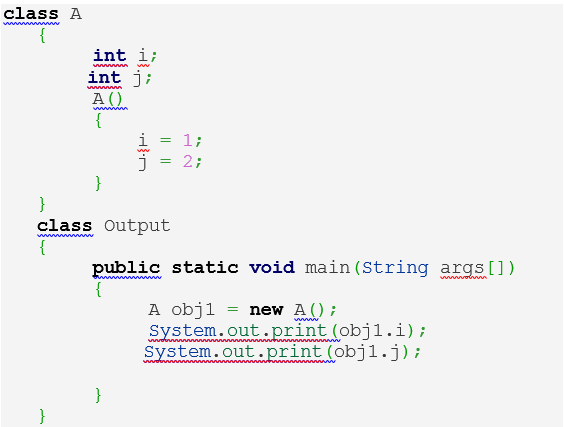
* **Name of the depositor**
* **Account Number**
* **Type of Account**
* **Balance amount**

**Methods:**

* **To assign initial values**
* **To deposit an amount**
* **To withdraw an amount after checking balance**
* **To display the name and balance**

**Task 7: Modify task 5 to use concept of array of objects.**

**Task 8: What will be the output of the following Java Code?**

****

**Task 9: Modify task 6 to incorporate concept of constructor to provide initial values to the objects.**

**Task 10: Write a Java program to find area of below shapes (use concept of method overloading)**

1. **Square**
2. **Rectangle**
3. **Circle**
4. **Triangle**

**A.2 Prerequisite:**

1. Fundamental concepts of C\C++.

2. Control statements in Java

**A.3 Outcome:**

**After successful completion of this experiment, students will be able to**

1. Implement concept of classes and objects.
2. Implement concept of methods, constructors and overloading of methods

**A.4 Theory:**

**A.4.1.**

## **Classes in Java**

A class is a blueprint from which individual objects are created.

Following is a sample of a class.

### **Example**

public class Dog {

String breed;

int age;

String color;

void barking() {

}

void hungry() {

}

void sleeping() {

}

}

A class can contain any of the following variable types.

* **Local variables** − Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.
* **Instance variables** − Instance variables are variables within a class but outside any method. These variables are initialized when the class is instantiated. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.
* **Class variables** − Class variables are variables declared within a class, outside any method, with the static keyword.

## **Objects in Java**

Let us now look deep into what are objects. If we consider the real-world, we can find many objects around us, cars, dogs, humans, etc. All these objects have a state and a behavior.

If we consider a dog, then its state is - name, breed, color, and the behavior is - barking, wagging the tail, running.

If you compare the software object with a real-world object, they have very similar characteristics.

Software objects also have a state and a behavior. A software object's state is stored in fields and behavior is shown via methods.

So in software development, methods operate on the internal state of an object and the object-to-object communication is done via methods.

## **Constructors**

When discussing about classes, one of the most important sub topic would be constructors. Every class has a constructor. If we do not explicitly write a constructor for a class, the Java compiler builds a default constructor for that class.

Each time a new object is created, at least one constructor will be invoked. The main rule of constructors is that they should have the same name as the class. A class can have more than one constructor.

Following is an example of a constructor −

### **Example**

public class Puppy {

public Puppy() {

}

public Puppy(String name) {

// This constructor has one parameter, *name*.

}

}

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the portal at the end of the practical. The filename should be B**DS\_batch\_rollno\_experimentno Example: BTI\_BDS\_C1\_C001\_Exp1**

|  |  |
| --- | --- |
| Roll No. A073 | Name: Aryan Srivastava |
| Class : MCA FY | Batch : B3 |
| Date of Experiment: 05-08-2024 | Date of Submission |
| Grade : |  |

**B.1 Software Code written by student:**

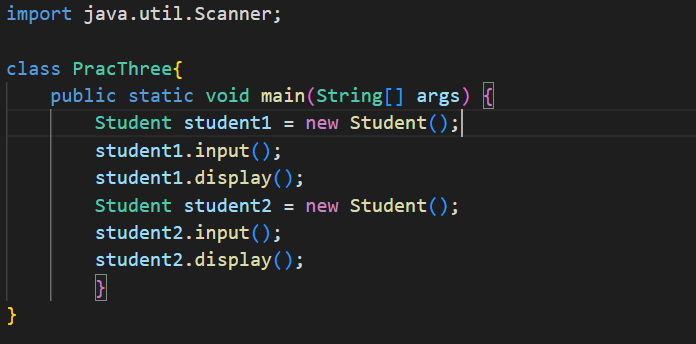
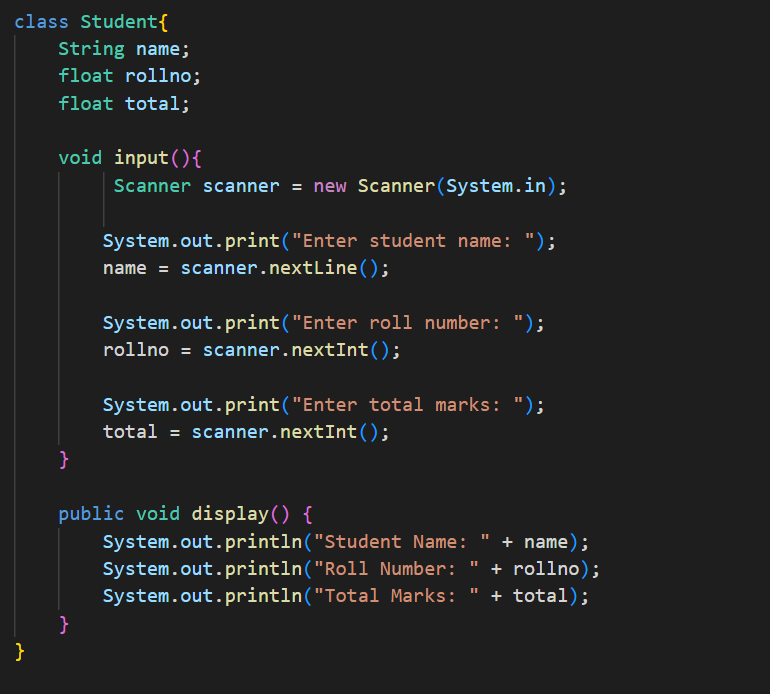
**Task 1: a. data**

**Task 2: c. reference**

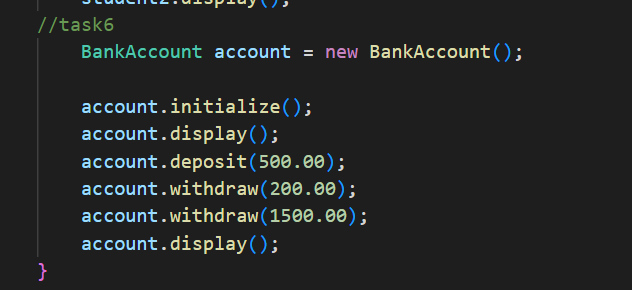
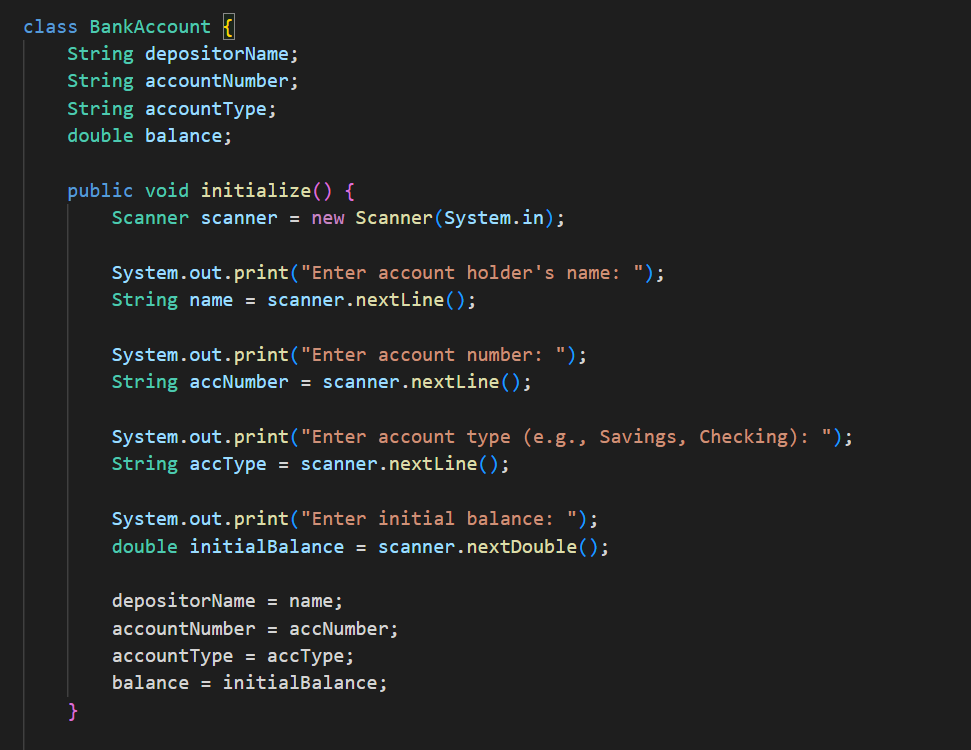
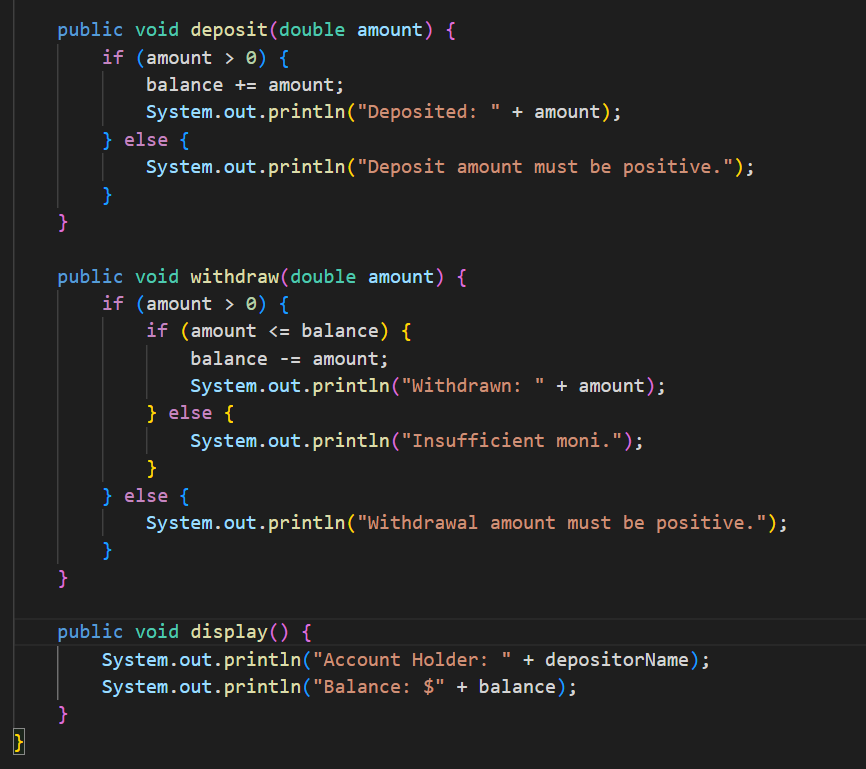
**Task 3: b. Data**

**Task 4: OP: 1**

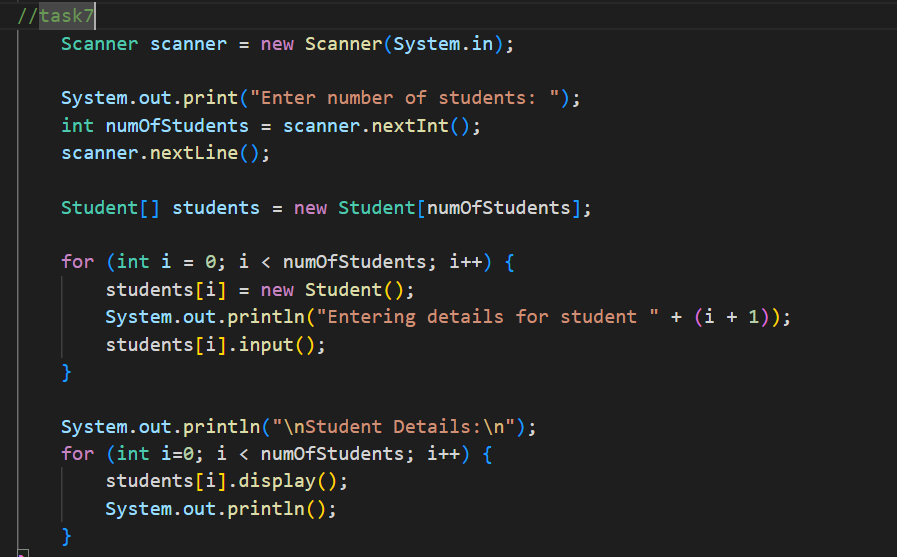
**Task 5:**

**** ****

**Task 6:**

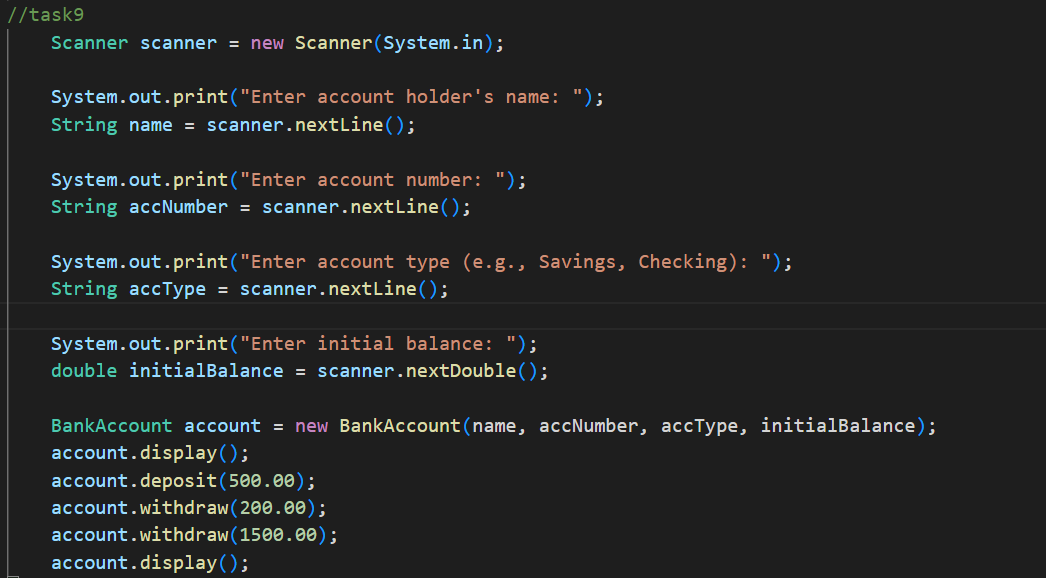
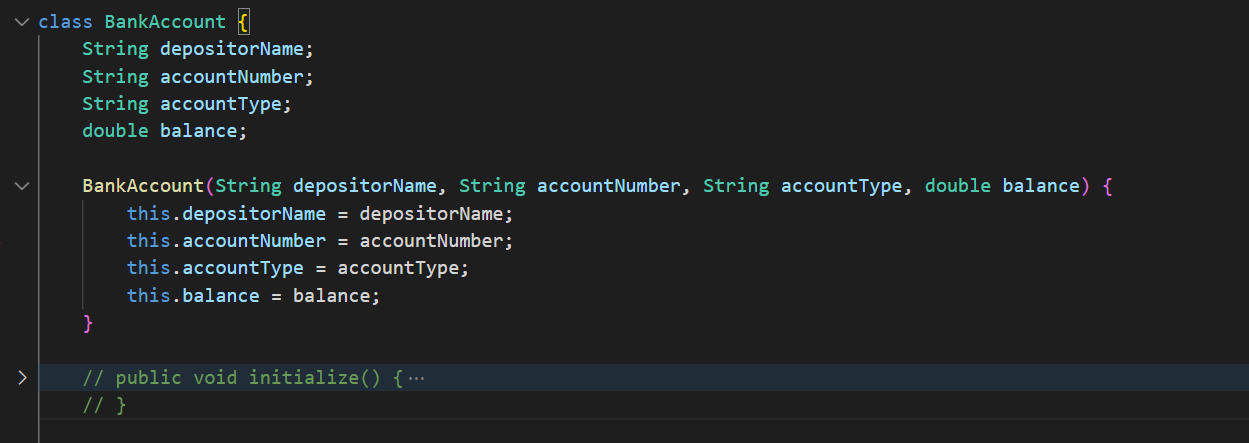
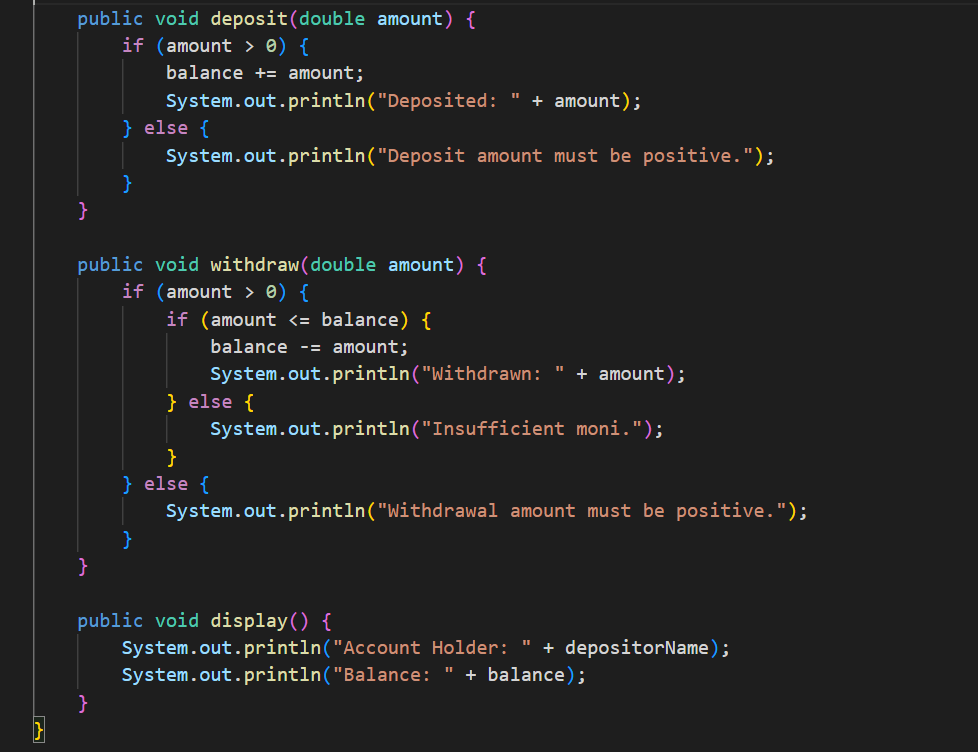
**** ---------->main()**** ****

**Task 7:**

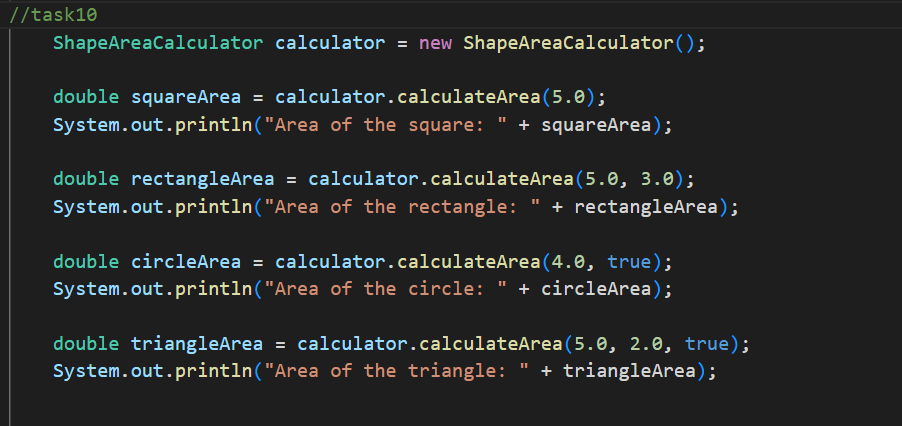
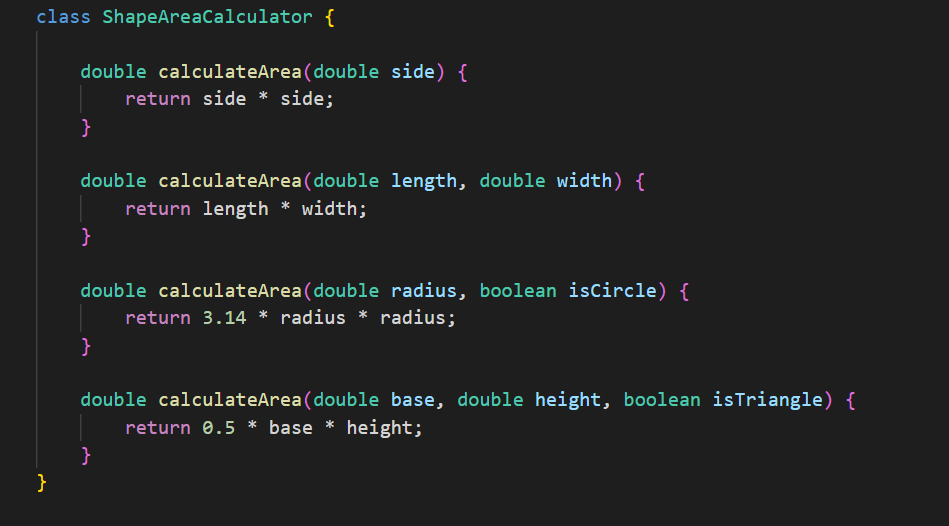
****

**Task 8: 12**

**Task 9:**

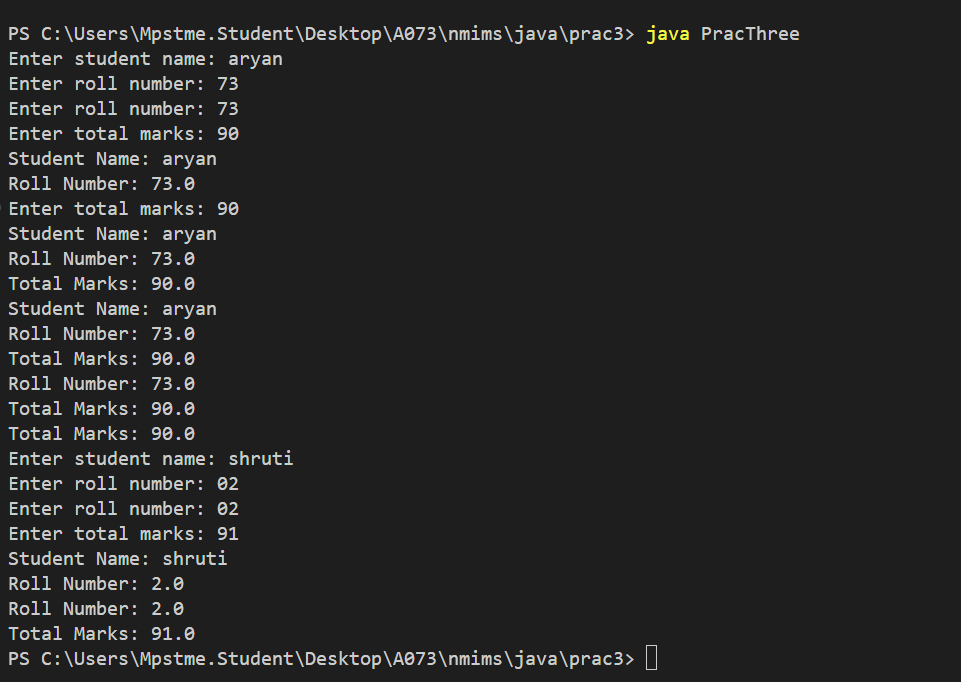
**Task 10:**

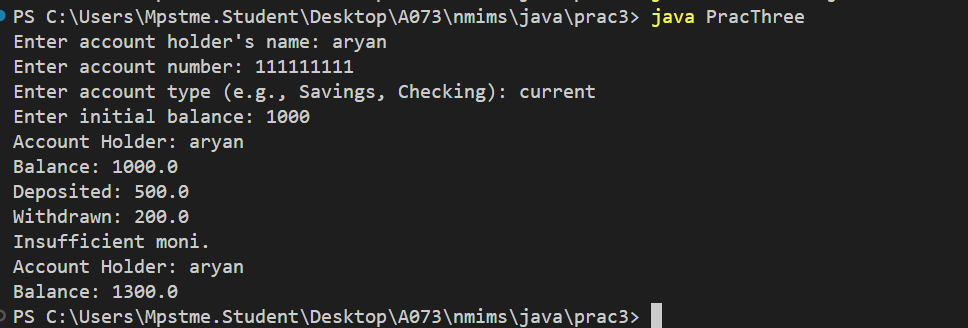
**** ****

**B.2 Input and Output:**

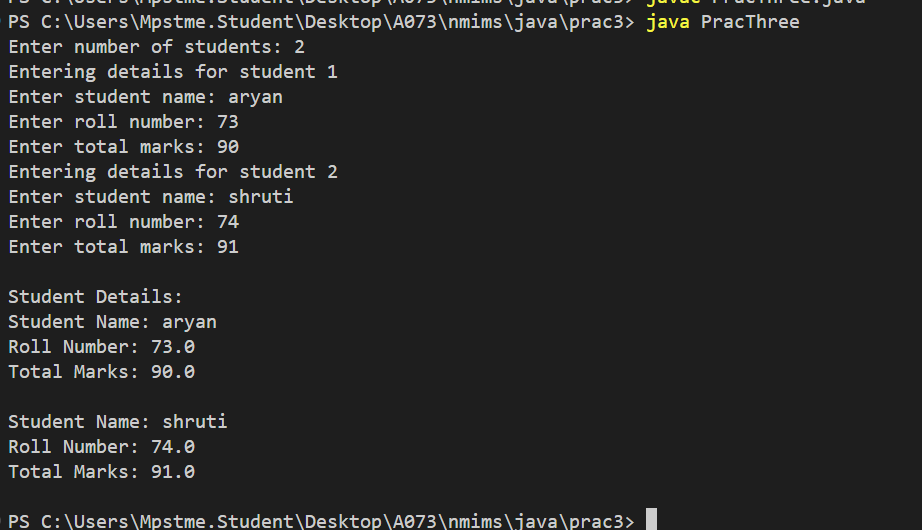
***(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

**Task 5:**

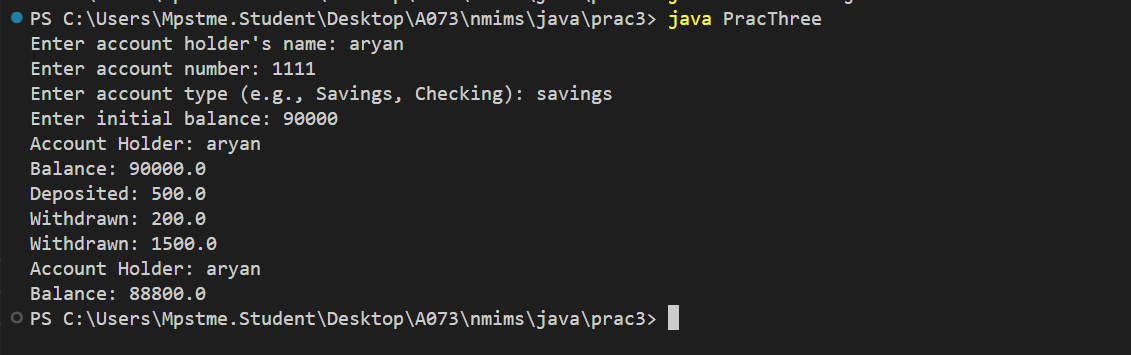
****

**Task 6:** ****

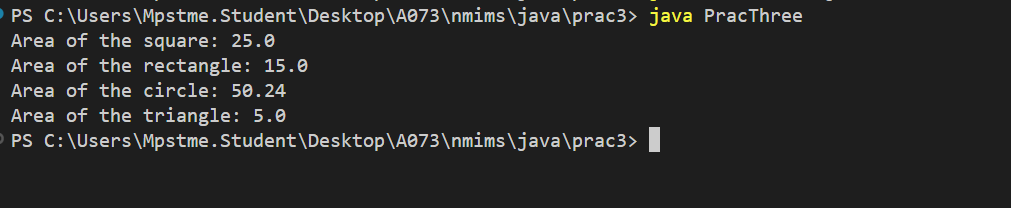
**Task 7:**

****

**Task 9:**

****

**Task 10:**

****

**B.3 Question of Curiosity:**

**Q.1 How do classes help us to organize our programs?**

**Q.2 What is a constructor? What are its special properties?**

**Q.3 How do we invoke constructor?**

**B.3 Conclusion:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*