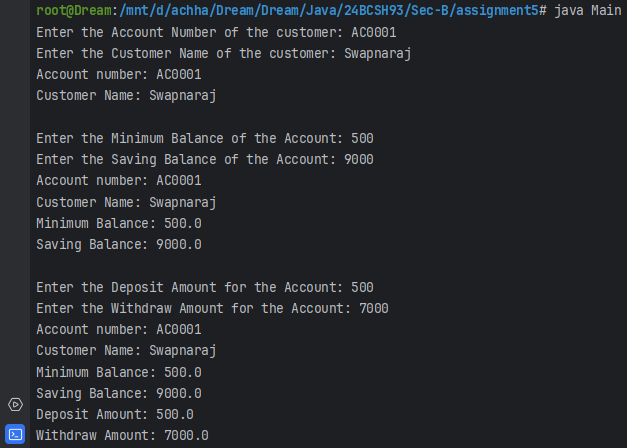
**Assignment – 5**

**Q1. Write a program that creates a class Account that have members accountNumber, customerName. Account has constructor to initialize its members and method display() to show the result. Create a child class SavingsAccount that is derived from Account class. SavingsAccount have members minBalance and savingBalance. Use show() to display its details. Then create another child class AccountDetails from SavingsAccount class have members deposit, withdraw and a method show1() to show its details. Child classes have constructors of their own. Create a Main class that creates a record of customer and display all its details.**

import java.util.Scanner;  
class Account {  
 private String accountNumber, customerName;  
  
 public Account(String accountNumber, String customerName) {  
 this.accountNumber = accountNumber;  
 this.customerName = customerName;  
 }  
  
 public void display() {  
 System.*out*.println("Account number: " + accountNumber);  
 System.*out*.println("Customer Name: " + customerName);  
 }  
  
 protected static void validate(String name, double value) {  
 if (value < 0) {  
 System.*out*.println(name + " cannot be negative");  
 System.*exit*(0);  
 }  
 }  
}  
  
class SavingsAccount extends Account {  
 private double minBalance, savingBalance;  
  
 public SavingsAccount(String accountNumber, String customerName, double minBalance, double savingBalance) {  
 super(accountNumber, customerName);  
 *validate*("Minimum Balance", minBalance);  
  
 if (savingBalance < minBalance) {  
 System.*out*.println("Saving Balance cannot be less than Minimum Balance");  
 }  
 this.minBalance = minBalance;  
 this.savingBalance = savingBalance;  
 }  
  
 public void show() {  
 display();  
 System.*out*.println("Minimum Balance: " + minBalance);  
 System.*out*.println("Saving Balance: " + savingBalance);  
 }  
}

class AccountDetails extends SavingsAccount {  
 private double deposit, withdraw;  
  
 public AccountDetails(String accountNumber, String name, double minBalance, double savingBalance, double deposit, double withdraw) {  
 super(accountNumber, name, minBalance, savingBalance);  
 *validate*("Deposit Amount", deposit);  
 *validate*("Withdraw Amount", withdraw);  
 this.deposit = deposit;  
 this.withdraw = withdraw;  
 }  
  
 public void show1() {  
 show();  
 System.*out*.println("Deposit Amount: " + deposit);  
 System.*out*.println("Withdraw Amount: " + withdraw);  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the Account Number of the customer: ");  
 String accountNumber = sc.next();  
 System.*out*.print("Enter the Customer Name of the customer: ");  
 String name = sc.next();  
  
 Account account = new Account(accountNumber, name);  
 account.display();  
  
 System.*out*.print("\nEnter the Minimum Balance of the Account: ");  
 double minBalance = sc.nextDouble();  
 System.*out*.print("Enter the Saving Balance of the Account: ");  
 double savingBalance = sc.nextDouble();  
  
 SavingsAccount savingsAccount = new SavingsAccount(accountNumber, name, minBalance, savingBalance);  
 savingsAccount.show();  
  
 System.*out*.print("\nEnter the Deposit Amount for the Account: ");  
 double deposit = sc.nextDouble();  
 System.*out*.print("Enter the Withdraw Amount for the Account: ");  
 double withdraw = sc.nextDouble();  
  
 AccountDetails accountDetails = new AccountDetails(accountNumber, name, minBalance, savingBalance, deposit, withdraw);  
 accountDetails.show1();  
 }  
}

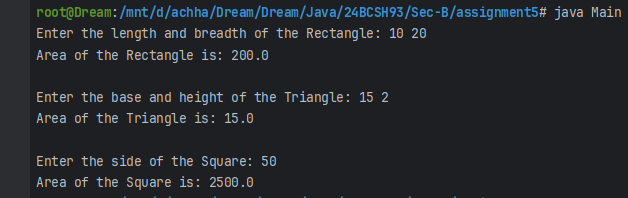
**Output:**

****

**Q2. Create a class Figure with instance members dim1 and dim2. Use constructor and area() that returns the area of figure. Create a derived class Rectangle derived from Figure and area() that returns the area of rectangle. Create another derived class Triangle and Square that has area() which returns area of Triangle and Square respectively. Derived class have appropriate constructor. Using method overriding concept test the functionalities of derived class by creating objects and super class memory references in Main class.**

import java.util.Scanner;  
abstract class Figure {  
 protected double dim1, dim2;  
  
 public Figure(double dim1, double dim2) {  
 *validate*("First Dimension", dim1);  
 *validate*("Second Dimension", dim2);  
 this.dim1 = dim1;  
 this.dim2 = dim2;  
 }  
  
 protected static void validate(String name, double value) {  
 if (value <= 0.0) {  
 System.*out*.println(name + " can only be positive");  
 }  
 }  
  
 abstract double area();  
}  
  
class Rectangle extends Figure {  
 public Rectangle(double length, double breadth) {  
 super(length, breadth);  
 }  
  
 public double area() {  
 return dim1 \* dim2;  
 }  
}  
  
class Triangle extends Figure {  
 public Triangle(double base, double height) {  
 super(base, height);  
 }  
  
 public double area() {  
 return 0.5 \* dim1 \* dim2;  
 }  
}  
  
class Square extends Figure {  
 public Square(double side) {  
 super(side, side);  
 }  
  
 public double area() {  
 return dim1 \* dim1;  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 Figure figure;  
  
 System.*out*.print("Enter the length and breadth of the Rectangle: ");  
 double length = sc.nextDouble();  
 double breadth = sc.nextDouble();  
 figure = new Rectangle(length, breadth);  
 System.*out*.println("Area of the Rectangle is: " + figure.area());  
  
 System.*out*.print("\nEnter the base and height of the Triangle: ");  
 double base = sc.nextDouble();  
 double height = sc.nextDouble();  
 figure = new Triangle(base, height);  
 System.*out*.println("Area of the Triangle is: " + figure.area());  
  
 System.*out*.print("\nEnter the side of the Square: ");  
 double side = sc.nextDouble();  
 figure = new Square(side);  
 System.*out*.println("Area of the Square is: " + figure.area());  
 }  
}

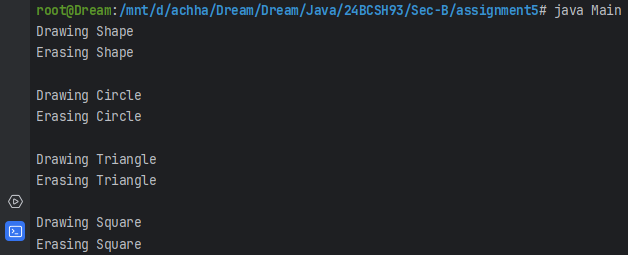
**Output:**



**Q3. Write a program to create a class named Shape. It should contain two methods, draw() and erase() that prints “Drawing Shape” and “Erasing Shape” respectively. For this class, create three sub classes, Circle, Triangle and Square and each class should override the parent class functions- draw() and erase(). The draw() method should print “Drawing Circle”, “Drawing Triangle” and “Drawing Square” respectively. The erase() method should print “Erasing Circle”, “Erasing Triangle” and “Erasing Square” respectively. Create objects of Circle, Triangle and Square, assign each to Shape variable(reference) and call draw() and erase() method using each object.**

class Shape {  
 public void draw() {  
 System.*out*.println("Drawing Shape");  
 }  
  
 public void erase() {  
 System.*out*.println("Erasing Shape");  
 }  
}  
  
class Circle extends Shape {  
 public void draw() {  
 System.*out*.println("Drawing Circle");  
 }  
  
 public void erase() {  
 System.*out*.println("Erasing Circle");  
 }  
}  
  
class Triangle extends Shape {  
 public void draw() {  
 System.*out*.println("Drawing Triangle");  
 }  
  
 public void erase() {  
 System.*out*.println("Erasing Triangle");  
 }  
}  
  
class Square extends Shape {  
 public void draw() {  
 System.*out*.println("Drawing Square");  
 }  
  
 public void erase() {  
 System.*out*.println("Erasing Square");  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Shape shape = new Shape();  
 shape.draw();  
 shape.erase();  
 System.*out*.println();  
 shape = new Circle();  
 shape.draw();  
 shape.erase();  
 System.*out*.println();  
  
 shape = new Triangle();  
 shape.draw();  
 shape.erase();  
 System.*out*.println();  
  
 shape = new Square();  
 shape.draw();  
 shape.erase();  
 }  
}

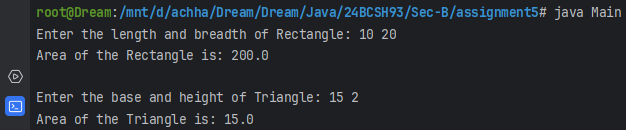
**Output:**



**Q4. Define an abstract class named “Figure”, having data members dim1 and dim2. Extend this class to create two concrete classes named Rectangle and Triangle. Override the getArea() method in the sub classes. Invoke the getArea() method in the main method of another Main class through the abstract class reference variable.**

import java.util.Scanner;  
abstract class Figure {  
 protected double dim1, dim2;  
  
 protected static void validate(String name, double value) {  
 if (value <= 0.0) {  
 System.*out*.println(name + " can only be positive");  
 }  
 }  
  
 abstract double getArea();  
}  
  
class Rectangle extends Figure {  
 public Rectangle(double length, double breadth) {  
 *validate*("Length", length);  
 *validate*("Breadth", breadth);  
 this.dim1 = length;  
 this.dim2 = breadth;  
 }  
  
 public double getArea() {  
 return dim1 \* dim2;  
 }  
}  
  
class Triangle extends Figure {  
 public Triangle(double base, double height) {  
 *validate*("Base", base);  
 *validate*("Height", height);  
 this.dim1 = base;  
 this.dim2 = height;  
 }  
  
 public double getArea() {  
 return 0.5 \* dim1 \* dim2;  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 Figure figure;  
  
 System.*out*.print("Enter the length and breadth of Rectangle: ");  
 double length = sc.nextDouble();  
 double breadth = sc.nextDouble();  
 figure = new Rectangle(length, breadth);  
 System.*out*.println("Area of the Rectangle is: " + figure.getArea());  
  
 System.*out*.print("\nEnter the base and height of Triangle: ");  
 double base = sc.nextDouble();  
 double height = sc.nextDouble();  
 figure = new Triangle(base, height);  
 System.*out*.println("Area of the Triangle is: " + figure.getArea());  
 }  
}

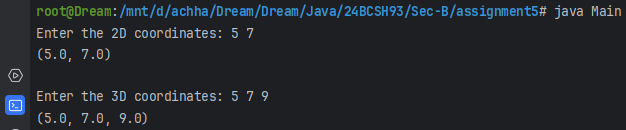
**Output:**



**Q5. Create a class Point2D with the data member x and y coordinate. Use default and parameterised constructor to set the coordinate values and display() to show the coordinates. Create a subclass called Point3D which is derived from the superclass Point2D with data members z coordinate and has constructor to initialize the input and show() method to display the coordinates. Test the methods of both the classes by creating objects in the main method of driver class.**

import java.util.Scanner;  
class Point2D {  
 protected double x, y;  
   
 public Point2D() {  
 x = y = 0.0;  
 }  
   
 public Point2D(double x, double y) {  
 this.x = x;  
 this.y = y;  
 }  
   
 public void display() {  
 System.*out*.println("(" + x + ", " + y + ")");  
 }  
}  
  
class Point3D extends Point2D {  
 private double z;  
   
 public Point3D() {  
 super();  
 z = 0.0;  
 }  
   
 public Point3D(double x, double y, double z) {  
 super(x, y);  
 this.z = z;  
 }  
   
 public void show() {  
 System.*out*.println("(" + x + ", " + y + ", " + z + ")");  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the 2D coordinates: ");  
 double x = sc.nextDouble();  
 double y = sc.nextDouble();  
 Point2D point2D = new Point2D(x, y);  
 point2D.display();  
   
 System.*out*.print("\nEnter the 3D coordinates: ");  
 x = sc.nextDouble();  
 y = sc.nextDouble();  
 double z = sc.nextDouble();  
 Point3D point3D = new Point3D(x, y, z);  
 point3D.show();  
 }  
}

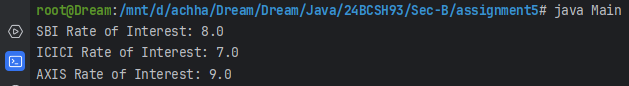
**Output:**



**Q6. Create the classes as given in the below figure. Display the interest rate in the following format:  
 SBI Rate of Interest : 8  
 ICICI Rate of Interest : 7  
 AXIS Rate of Interest : 9**

abstract class Bank {  
 abstract double getRateOfInterest();  
}  
  
class SBI extends Bank {  
 public double getRateOfInterest() {  
 return 8.0;  
 }  
}  
  
class ICICI extends Bank {  
 public double getRateOfInterest() {  
 return 7.0;  
 }  
}  
  
class AXIS extends Bank {  
 public double getRateOfInterest() {  
 return 9.0;  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Bank bank;  
 bank = new SBI();  
 System.*out*.println("SBI Rate of Interest: " + bank.getRateOfInterest());  
 bank = new ICICI();  
 System.*out*.println("ICICI Rate of Interest: " + bank.getRateOfInterest());  
 bank = new AXIS();  
 System.*out*.println("AXIS Rate of Interest: " + bank.getRateOfInterest());  
 }  
}

**Output:**



**Q7.** **Create a class Person that has data member name. Use constructor to initialize name and display() to display name. Create a derived class Employee from Person class having private members id. Using constructor initialize id and have method display() to display id. Create another derived class HourlyEmployee from Employee with private members hourlyRate and hoursWorked. Use constructor to initialize input and methods getGrossPay() that computes and returns the gross pay of the employee and display() to display the hourlyRate, hoursWorked and gross pay. Create a Main class to test the functionalities of the above classes and display output in the following format  
Name: John Smith  
ID: 7569  
Hourly Rate: 100  
Hours Worked: 2000  
Gross pay: 200000**

import java.util.Scanner;  
class Person {  
 private String name;  
  
 public Person(String name) {  
 this.name = name;  
 }  
  
 public void display() {  
 System.*out*.println("Name: " + name);  
 }  
}  
  
class Employee extends Person {  
 private int id;  
  
 public Employee(String name, int id) {  
 super(name);  
 *validate*("ID", id);  
 this.id = id;  
 }  
  
 protected static void validate(String name, double value) {  
 if (value <= 0.0) {  
 System.*out*.println(name + " can only be positive");  
 System.*exit*(0);  
 }  
 }  
  
 public void display() {  
 super.display();  
 System.*out*.println("ID: " + id);  
 }  
}

class HourlyEmployee extends Employee {  
 private double hourlyRate;  
 private int hoursWorked;  
  
 public HourlyEmployee(String name, int id, double hourlyRate, int hoursWorked) {  
 super(name, id);  
 *validate*("Hourly Rate", hourlyRate);  
 *validate*("Hours Worked", hoursWorked);  
 this.hourlyRate = hourlyRate;  
 this.hoursWorked = hoursWorked;  
 }  
  
 public double getGrossPay() {  
 return hoursWorked \* hourlyRate;  
 }  
  
 public void display() {  
 super.display();  
 System.*out*.println("Hourly Rate: " + hourlyRate);  
 System.*out*.println("Hours Worked: " + hoursWorked);  
 System.*out*.println("Gross Pay: " + getGrossPay());  
 }  
}  
  
class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter the Name of the Person: ");  
 String name = sc.nextLine();  
  
 Person person = new Person(name);  
 person.display();  
 System.*out*.print("\nEnter ID of the Employee: ");  
 int id = sc.nextInt();  
  
 person = new Employee(name, id);  
 person.display();  
  
 System.*out*.print("\nEnter the Hourly Rate of the HourlyEmployee: ");  
 double hourlyRate = sc.nextDouble();  
 System.*out*.print("Enter the Hours Worked of the HourlyEmployee: ");  
 int hoursWorked = sc.nextInt();  
  
 person = new HourlyEmployee(name, id, hourlyRate, hoursWorked);  
 person.display();  
 }  
}

**Output:**

