1.Create an abstract class named Shape with the following protected attributes / member variables.

String name

Include a 1-argument constructor.

Include getters and setters.

Include an abstract method named calculateArea() . This method returns a Float value.

Create a class named Circle . The class Circle is a derived class of Shape. Include the following private attributes / member variables.

Integer radius

Include a 2-argument constructor. The order of the arguments is name, radius.

Include getters and setters.

Override the abstract method calculateArea() defined in the Shape class. This method returns the area of the circle. [Take the value of pi as 3.14]

Create a class named Square . The class Square is a derived class of Shape. Include the following private attributes / member variables.

Integer side

Include a 2-argument constructor. The order of the arguments is name, side.

Include getters and setters.

Override the abstract method calculateArea() defined in the Shape class. This method returns the area of the square.

Create a class named Rectangle . The class Rectangle is a derived class of Shape. Include the following private attributes / member variables.

Integer length

Integer breadth

Include a 3-argument constructor. The order of the arguments is name, length, breadth

Include getters and setters.

Override the abstract method calculateArea() defined in the Shape class. This method returns the area of the rectangle.

Create another class called Main. In the method, create instances of the above classes and test the above classes.

All Float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

Sample Input and Output 1:

Circle

Square

Rectangle

Enter the shape name

Circle

Enter the radius

25

Area of Circle is 1962.50

2. A Java interface can only contain method signatures and fields. The interface can be used to achieve polymorphism. In this problem, you will practice your knowledge on interfaces.

You are given an interface AdvancedArithmetic which contains a method signature int divisor\_sum(int n). You need to write a class called MyCalculator which implements the interface.

divisorSum function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so divisor\_sum should return 12. The value of n will be at most 1000.

Read the partially completed code in the editor and complete it. You just need to write the MyCalculator class only. Your class shouldn't be public.

Sample Input

6

Sample Output

I implemented: AdvancedArithmetic

12

Explanation

Divisors of 6 are 1,2,3 and 6. 1+2+3+6=12.

3.Create an abstract class named Card with the following protected attributes / member variables.

String holderName;

String cardNumber;

String expiryDate;

Include appropriate getters and setters.

Include appropriate constructors. In the 3-argument constructor, the order of the arguments is holderName, cardNumber, expiryDate.

Create a class named MembershipCard. The class MembershipCard is a derived class of Card. Include the following private attributes / member variables.

Integer rating

Include appropriate getters and setters.

Include appropriate constructors. In the 4-argument constructor, the order of the arguments is holderName, cardNumber, expiryDate, rating.

Create a class named PaybackCard. The class PaybackCard is a derived class of Card. Include the following private attributes / member variables.

Integer pointsEarned;

Double totalAmount;

Include appropriate getters and setters.

Include appropriate constructors. In the 5-argument constructor, the order of the arguments is holderName, cardNumber, expiryDate, pointsEarned, totalAmount.

Create another class called Main. In the method, create instances of the above classes and test the above classes. The card details are entered separated by a ‘|’.

All text in bold corresponds to input and the rest corresponds to output.

Sample Input and Output 1:

Select the Card

1.Payback Card

2.Membership Card

1

Enter the Card Details:

Anandhi|12345|14/01/2020

Enter points in card

1000

Enter Amount

50000

Anandhi's Payback Card Details:

Card Number 12345

Points Earned 1000

Total Amount 50000.0

4. The following Java applications contain errors. Point out the statement(s) that contain errors. Explain what each of the errors is, and how it can be fixed.

public class OOPExercises { public static void main(String[] args) { A objA = new A(); System.out.println("in main(): "); System.out.println("objA.a = "+objA.a); objA.a = 222; } }

public class A {

private int a = 100;

public void setA( int value) {

a = value;

}

public int getA() {

return a;

}

} //class A

5. Show the output of the following applications.

public class OOPExercises { public static void main(String[] args) { FirstClass objA = new FirstClass(); SecondClass objB = new SecondClass(); System.out.println("in main(): "); System.out.println("objA.a = "+objA.getFirstClass()); System.out.println("objB.b = "+objB.getSecondClass()); objA.setFirstClass (222); objB.setSecondClass (333.33); System.out.println("objA.a = "+objA.getFirstClass()); System.out.println("objB.b = "+objB.getSecondClass()); } } Output:

public class FirstClass {

int a = 100; public FirstClass () { System.out.println("in the constructor of class FirstClass: "); System.out.println("a = "+a); a = 333; System.out.println("a = "+a); } public void setFirstClass( int value) { a = value; } public int getFirstClass() { return a; } } //class FirstClass

public class SecondClass {

double b = 123.45;

public SecondClass() {

System.out.println("-----in the constructor of class B: ");

System.out.println("b = "+b);

b = 3.14159;

System.out.println("b = "+b);

}

public void setSecondClass( double value) {

b = value;

}

public double getSecondClass() {

return b;

}

} //class SecondClass

6.

public class OOPExercises { static int a = 555; public static void main(String[] args) { A objA = new A(); B objB1 = new B(); A objB2 = new B(); C objC1 = new C(); B objC2 = new C(); A objC3 = new C(); objA.display(); objB1.display(); objB2.display(); objC1.display(); objC2.display(); objC3.display(); } } Output:

public class A {

int a = 100;

public void display() {

System.out.printf("a in A = %d\n", a);

}

} //class A

public class B extends A {

private int a = 123;

public void display() {

System.out.printf("a in B = %d\n", a);

}

} //class B

public class C extends B {

private int a = 543;

public void display() {

System.out.printf("a in