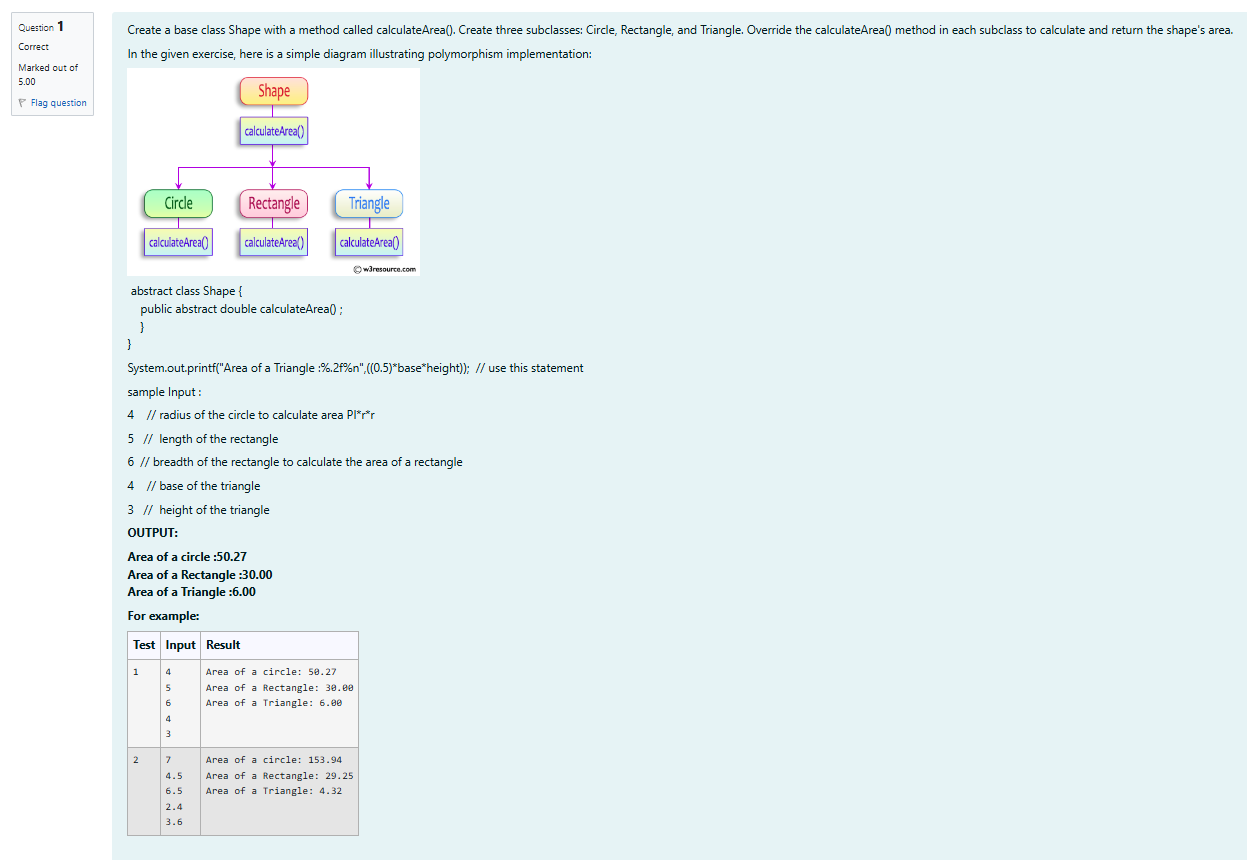
### **CS23333-Object Oriented Programming Using Java**

**NAME: B. ELUMALAI**

**ROLL NO: 230701084**

## **Lab-08-Logic Building**



**CODE**

import java.util.Scanner;

// Abstract class Shape

abstract class Shape {

public abstract double calculateArea();

}

// Circle class

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double calculateArea() {

return Math.PI \* radius \* radius; // Area of circle: πr²

}

}

// Rectangle class

class Rectangle extends Shape {

private double length;

private double breadth;

public Rectangle(double length, double breadth) {

this.length = length;

this.breadth = breadth;

}

@Override

public double calculateArea() {

return length \* breadth; // Area of rectangle: length \* breadth

}

}

// Triangle class

class Triangle extends Shape {

private double base;

private double height;

public Triangle(double base, double height) {

this.base = base;

this.height = height;

}

@Override

public double calculateArea() {

return 0.5 \* base \* height; // Area of triangle: 0.5 \* base \* height

}

}

// Main class to test the shapes

public class ShapeTest {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input for Circle

double radius = scanner.nextDouble();

Circle circle = new Circle(radius);

System.out.printf("Area of a circle: %.2f%n", circle.calculateArea());

// Input for Rectangle

double length = scanner.nextDouble();

double breadth = scanner.nextDouble();

Rectangle rectangle = new Rectangle(length, breadth);

System.out.printf("Area of a Rectangle: %.2f%n", rectangle.calculateArea());

// Input for Triangle

double base = scanner.nextDouble();

double height = scanner.nextDouble();

Triangle triangle = new Triangle(base, height);

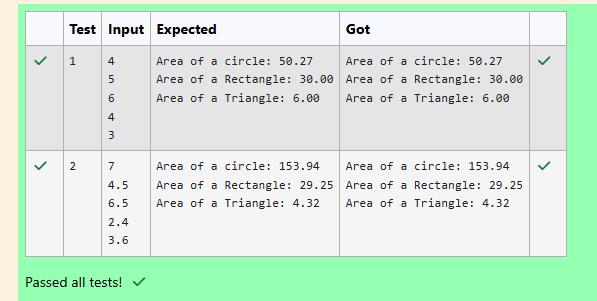
System.out.printf("Area of a Triangle: %.2f%n", triangle.calculateArea());

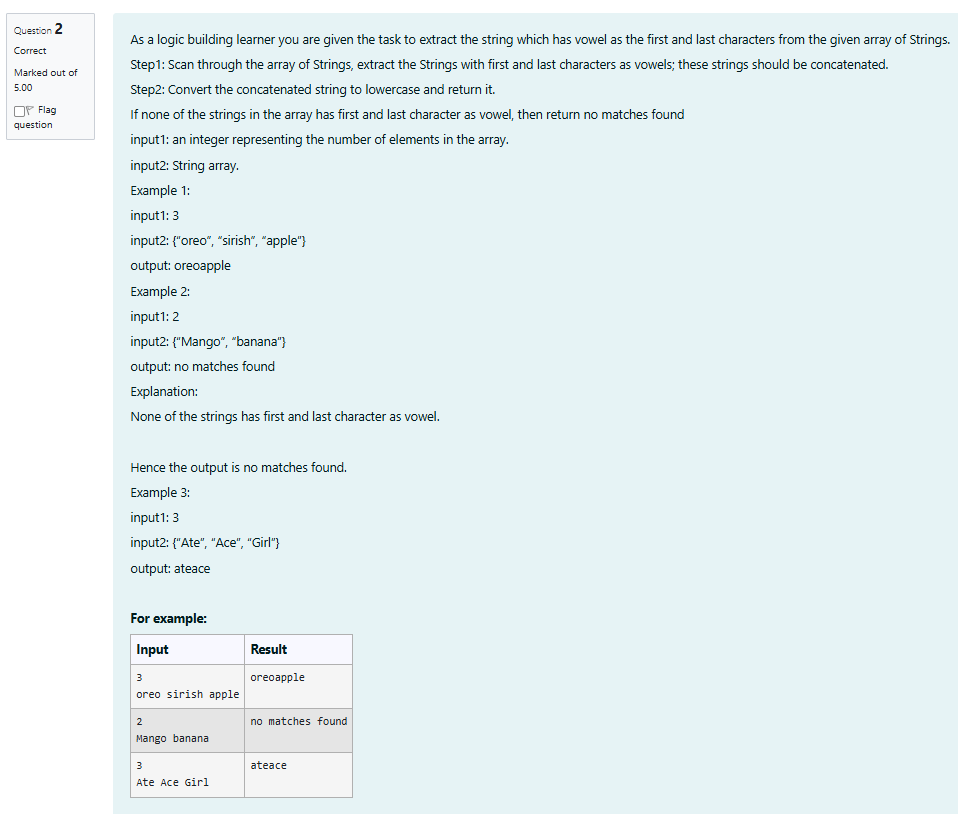
scanner.close();

}

}

**OUTPUT**

****

****

**CODE**

import java.util.Scanner;

public class VowelStringExtractor {

// Method to extract strings with vowels as first and last characters

public static String extractVowelStrings(String[] stringArray) {

StringBuilder result = new StringBuilder();

String vowels = "aeiouAEIOU"; // String containing all vowels

// Iterate through the array of strings

for (String s : stringArray) {

// Check if the string is not empty and if both the first and last characters are vowels

if (s.length() > 0 && vowels.indexOf(s.charAt(0)) != -1 && vowels.indexOf(s.charAt(s.length() - 1)) != -1) {

result.append(s); // Append matching string to the result

}

}

// Return the concatenated string in lowercase or "no matches found"

return result.length() > 0 ? result.toString().toLowerCase() : "no matches found";

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input for the number of strings

int n = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Input for the strings in one line

String input = scanner.nextLine();

String[] strings = input.split(" "); // Split input into an array

// Process and output the result

String result = extractVowelStrings(strings);

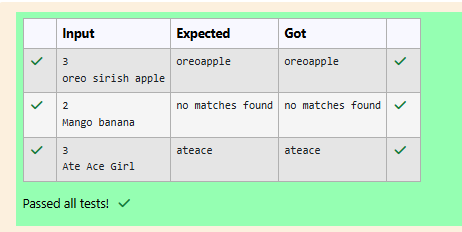
System.out.println(result);

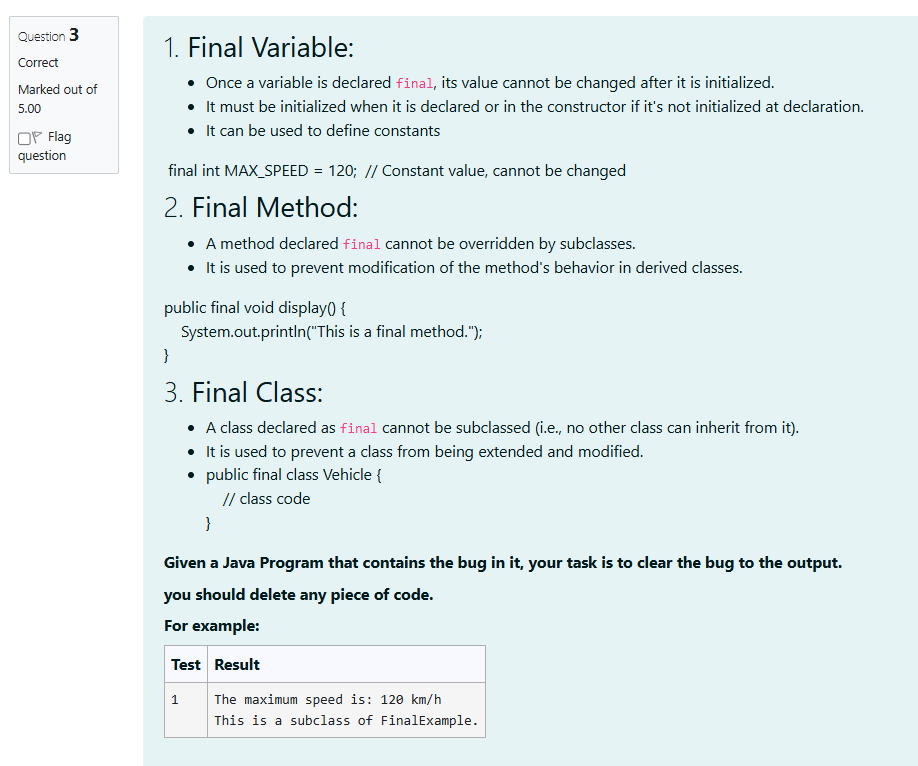
scanner.close(); // Close the scanner

}

}

**OUTPUT**

****

****

**CODE**

// Final class definition

final class FinalExample {

// Final variable

final int MAX\_SPEED = 120; // Constant value

// Final method

public final void display() {

System.out.println("The maximum speed is: " + MAX\_SPEED + " km/h");

}

}

// Main class to test the final class

public class Test {

public static void main(String[] args) {

// Create an instance of FinalExample

FinalExample example = new FinalExample();

example.display();

// Uncommenting the following line will result in a compile-time error

// because FinalExample is a final class and cannot be subclassed.

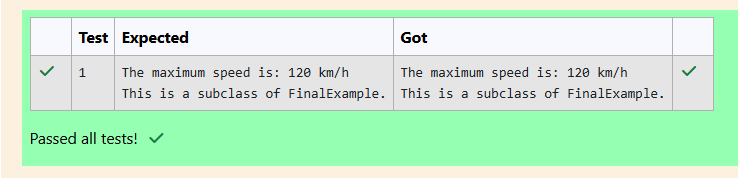
// class SubclassExample extends FinalExample { }

System.out.println("This is a subclass of FinalExample.");

}

}

**OUTPUT**

****