

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-08 - Polymorphism, Abstract Classes, final Keyword](#) / [Lab-08-Logic Building](#)

<b>Status</b>	Finished
<b>Started</b>	Thursday, 10 October 2024, 12:44 PM
<b>Completed</b>	Sunday, 13 October 2024, 12:17 PM
<b>Duration</b>	2 days 23 hours

## Question 1

Correct

Marked out of 5.00

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {"oreo", "sirish", "apple"}

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {"Ate", "Ace", "Girl"}

output: ateace

**For example:**

Input	Result
3 oreo sirish apple	oreoapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

**Answer:** (penalty regime: 0 %)

```

1 import java.util.*;
2 public class Main{
3     static StringBuilder checkVowel(String[] arr){
4         HashSet<Character> set = new HashSet<>();
5         char vowels[] = {'a','e','i','o','u'};
6         for(char c : vowels) set.add(c);
7         StringBuilder temp = new StringBuilder();
8         for(String str : arr){
9             if(set.contains(str.charAt(0)) && set.contains(str.charAt(str.length()-1))){
10                 temp.append(str);
11             }
12         }
13         if(temp.isEmpty()) temp.append("no matches found");
14         return temp;
15     }
16
17     public static void main(String []args){
18         Scanner scan = new Scanner(System.in);
19         int n = scan.nextInt();
20         String arr[] = new String[n];

```

	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2 Mango banana	no matches found	no matches found	✓
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓

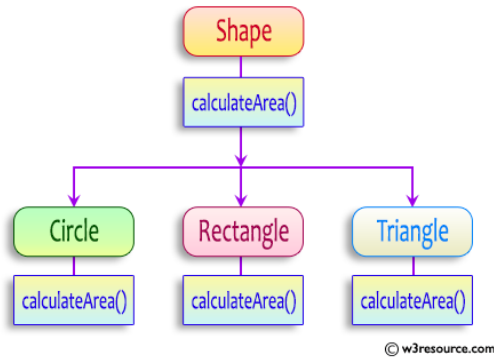
## Question 2

Correct

Marked out of 5.00

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:



```

abstract class Shape {
    public abstract double calculateArea() ;
}

```

```
System.out.printf("Area of a Triangle :%.2f%n",((0.5)*base*height)); // use this statement
```

sample Input :

```

4 // radius of the circle to calculate area PI*r*r
5 // length of the rectangle
6 // breadth of the rectangle to calculate the area of a rectangle
4 // base of the triangle
3 // height of the triangle

```

**OUTPUT:**

**Area of a circle :50.27**

**Area of a Rectangle :30.00**

**Area of a Triangle :6.00**

**For example:**

Test	Input	Result
1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00
2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32

**Answer:** (penalty regime: 0 %)

```

1 import java.util.*;
2 abstract class Shape{
3     public abstract double calculateArea();
4 }
5
6 abstract class Circle extends Shape{
7     public static double calculateArea(double radius){
8         return Math.PI * radius*radius;
9     }
10 }
11 abstract class Rectangle extends Shape{
12     public static double calculateArea(double length,double breadth){
13         return length * breadth;

```

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

## Question 3

Correct

Marked out of 5.00

## 1. Final Variable:

- Once a variable is declared **final**, its value cannot be changed after it is initialized.
- It must be initialized when it is declared or in the constructor if it's not initialized at declaration.
- It can be used to define constants

```
final int MAX_SPEED = 120; // Constant value, cannot be changed
```

## 2. Final Method:

- A method declared **final** cannot be overridden by subclasses.
- It is used to prevent modification of the method's behavior in derived classes.

```
public final void display() {  
    System.out.println("This is a final method.");  
}
```

## 3. Final Class:

- A class declared as **final** cannot be subclassed (i.e., no other class can inherit from it).
- It is used to prevent a class from being extended and modified.
- public final class Vehicle {  
 // class code  
}

Given a Java Program that contains the bug in it, your task is to clear the bug to the output.

you should delete any piece of code.

For example:

Test	Result
1	The maximum speed is: 120 km/h This is a subclass of FinalExample.

Answer: (penalty regime: 0 %)

Reset answer

```
1 class FinalExample {  
2  
3     // Final variable  
4     final int maxSpeed = 120;  
5  
6     // Final method  
7     public final void displayMaxSpeed() {  
8         System.out.println("The maximum speed is: " + maxSpeed + " km/h");  
9     }  
10 }  
11  
12 class SubClass extends FinalExample {  
13  
14     // public void displayMaxSpeed() {  
15     //     System.out.println("Cannot override a final method");  
16     // }  
17  
18     // You can create new methods here  
19     public void showDetails() {  
20         System.out.println("This is a subclass of FinalExample.");  
21     }  
22 }  
23  
24 class prog {  
25     public static void main(String[] args) {  
26         FinalExample obj = new FinalExample();  
27         obj.displayMaxSpeed();  
28  
29         SubClass subObj = new SubClass();  
30         subObj.showDetails();  
31     }  
32 }  
33 }
```

	Test	Expected	Got	
✓	1	The maximum speed is: 120 km/h This is a subclass of FinalExample.	The maximum speed is: 120 km/h This is a subclass of FinalExample.	✓

Passed all tests! ✓

◀ [Lab-08-MCQ](#)

Jump to...

[FindStringCode](#) ▶