



**NAME :- AASHIQ RAHAMAN**

**SEMESTER :- 5<sup>TH</sup>**

**SUBJECT :- OOP**

**SUBJECT CODE :- PCC-CS 504**

**DEPARTMENT :- COMPUTER SCIENCE &  
ENGINEERING**

**REGISTRATION NO :- 221000110175**

**ROLL NO :- 10000122020**

**DATE OF SUBMISSION: - 18 NOV 2024**

---

## ***INDEX***

SL. NO.	QUESTION OF PROGRAMMING FOR PROBLEM SOLVING	DATE OF EXPERIMENT	DATE OF SUBMISSION	TEACHER'S SIGNATURE
1.	<p>CREATE A CLASS TO REPRESENT A BANK ACCOUNT. THE BANK ACCOUNT HAS THE FOLLOWING MEMBERS: NAME OF DEPOSITOR, ACCOUNT NUMBER, TYPE OF ACCOUNT, INITIAL BALANCE IN ACCOUNT.</p> <p>THE BANK IS HAVING FOLLOWING METHODS TO ASSIGN INITIAL AMOUNT, TO DEPOSIT AN AMOUNT, TO WITHDRAW AN AMOUNT AFTER CHECKING THE BALANCE, TO DISPLAY THE NAME AND BALANCE.</p>	<i>NOV 2024</i>	<i>17<sup>TH</sup> NOV 2024</i>	
2.	<p>WRITE A PROGRAM TO COMPUTE PERIMETER OF CLASS TRIANGLE, RECTANGLE AND SQUARE USING PARAMETERIZED CONSTRUCTOR.</p>	<i>NOV 2024</i>	<i>17<sup>TH</sup> NOV 2024</i>	

## PROBLEM: - 1

Write a java class which consists of 5 integer data. Overload constructor to initialize those integer data members. Provide a method which sorts those integer data members using bubble sort.

### Solution:-

```
import java.util.Arrays;
class IntegerData {
    int[] numbers;
    public IntegerData(int a, int b, int c, int d, int e) {
        numbers = new int[] {a, b, c, d, e};
    }
    public void bubbleSort() {
        int n = numbers.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (numbers[j] > numbers[j + 1]) {
                    int temp = numbers[j];
                    numbers[j] = numbers[j + 1];
                    numbers[j + 1] = temp;
                }
            }
        }
    }
    public void printNumbers() {
        System.out.println("After sorting the Sorted Numbers: " + Arrays.toString(numbers));
    }
    public static void main(String[] args) {
        IntegerData data = new IntegerData(23, 5, 78, 2, 45);
        System.out.println("Before Sorting the numbers : " + Arrays.toString(data.numbers));
        data.bubbleSort();
        data.printNumbers();
    }
}
```

### AT TERMINAL:-

Before Sorting the numbers : [23, 5, 78, 2, 45]

After sorting the Sorted Numbers: [2, 5, 23, 45, 78]

**PROBLEM: - 2**

**Write a program to design a class Volume and then find out the volume of a Cube, Cylinder using method overloading.**

### Solution:-

```

class Volume {
    public double calculateVolume(double side) {
        return side * side * side;
    }
    public double calculateVolume(double radius, double height) {
        return Math.PI * radius * radius * height;
    }
}

public class Main {
    public static void main(String[] args) {
        Volume volume = new Volume();
        double cubeVolume = volume.calculateVolume(5.0);
        System.out.println("Volume of Cube: " + cubeVolume);
        double cylinderVolume = volume.calculateVolume(3.0, 7.0);
        System.out.println("Volume of Cylinder: " + cylinderVolume);
    }
}

```

### AT TERMINAL:-

```
Volume of Cube: 125.0
Volume of Cylinder: 197.92033717615698
```

1. Java String Program to Check Anagram Input: str1 = "Listen" str2 = "Silent" Output: Yes An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

2. Program to Extract Substring from a String with Equal 0, 1, and 2. Sample Input: str = "102100211" Output: 5 Explanation: "102" , "021" , "210" , " 021" , "210021" these are combinations can be formed where the occurrence of 0 , 1 and 2 all are equal.

OPPS

## PROBLEM: - 1

Java String Program to Check Anagram Input: str1 = "Listen" str2 = "Silent" Output: Yes, an anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

And print the letters with its ASCII values.

### Solution:-

```
import java.util.Arrays;
public class AnagramCheck {
    public static void main(String[] args) {
        String str1 = "Listen";
        String str2 = "Silent";
        System.out.println("Before sorting the input strings:");
        System.out.println("str1: " + str1);
        System.out.println("str2: " + str2);
        if (areAnagrams(str1, str2)) {
            System.out.println("Yes, the strings are anagrams.");
        } else {
            System.out.println("No, the strings are not anagrams.");
        }
        asciiStr (str1, str2);
    }
    public static int asciiStr (String str1, String str2) {
        int astr1 = 0;
        int astr2 = 0;
        for (int i = 0; i < str1.length(); i++) {
            astr1 = (int) str1.charAt(i);
            System.out.print(str1.charAt(i)+" -> " + astr1+"; ");
        }
        System.out.println("");
        for (int i = 0; i < str2.length(); i++) {
            astr2 = (int) str2.charAt(i);
            System.out.print(str2.charAt(i)+ " -> " + astr2+"; ");
        }
        System.out.println("");
        return 0;
    }
    public static boolean areAnagrams(String str1, String str2) {
        str1 = str1.toLowerCase();
        str2 = str2.toLowerCase();
        if (str1.length() != str2.length()) {
            return false;
        }
        char[] charArray1 = str1.toCharArray();
        char[] charArray2 = str2.toCharArray();
        System.out.println("After Sorting the input strings accordingly:");
        Arrays.sort(charArray1);
```

```
        System.out.println(charArray1);  
        Arrays.sort(charArray2);  
        System.out.println(charArray2);  
        return Arrays.equals(charArray1, charArray2);  
    }  
}
```

## AT TERMINAL:-

Before sorting the input strings:

str1: Listen

str2: Silent

After Sorting the input strings accordingly:

eilnst

eilnst

Yes, the strings are anagrams.

L -> 76; i -> 105; s -> 115; t -> 116; e -> 101; n -> 110;

S -> 83; i -> 105; l -> 108; e -> 101; n -> 110; t -> 116;

## PROBLEM: - 2

Program to Extract Substring from a String with Equal 0, 1, and 2. Sample Input: str = "102100211" Output: 5 Explanation: "102" , "021" , "210" , " 021" , "210021" these are combinations can be formed where the occurrence of 0 , 1 and 2 all are equal.

### Solution:-

```
public class SubstringEqual012 {
    public static void main(String[] args) {
        String str = "102100211";
        System.out.println("Number of substrings with equal 0, 1, and 2: " +
countEqual012Substrings(str));
    }
    public static int countEqual012Substrings(String str) {
        int n = str.length();
        int count = 0;
        for (int i = 0; i < n; i++) {
            int count0 = 0, count1 = 0, count2 = 0;
            for (int j = i; j < n; j++) {
                char ch = str.charAt(j);
                if (ch == '0') count0++;
                else if (ch == '1') count1++;
                else if (ch == '2') count2++;
                if (count0 == count1 && count1 == count2) {
                    count++;
                }
            }
        }
        return count;
    }
}
```

### AT TERMINAL:-

Number of substrings with equal 0, 1, and 2: 5



**PROBLEM: - 1**

**Java String program to check whether a string is a Palindrome.**

**Solution:-**

```
import java.util.Scanner;
public class palin {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter a string to Check Wheather the stringis
palindrom or not: ");
        String input= scanner.nextLine();
        if (isPalindrome(input)) {
            System.out.println("Input String:\""+input+"\" is Palindrome");
        } else {
            System.out.println("Input String: \""+input+"\" Is Not Palindrome");
        }
    }
    public static boolean isPalindrome(String str) {
        str=str.toLowerCase();
        int left=0;
        int right=str.length()-1;
        while (left<right){
            if (str.charAt(left)!=str.charAt(right)){
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

**AT TERMINAL:-**

Enter a string to Check Wheather the stringis palindrom or not:

aba

Input String:"aba" is Palindrome

**Another input:**

Enter a string to Check Wheather the stringis palindrom or not:

Aashiq

Input String: "Aashiq" Is Not Palindrome

## PROBLEM: - 2

Java program to print the initials of your name.

Input: John David Smith

Output: J. D. S.

### Solution:-

```
import java.util.Scanner;
public class Initials{
    public static void main (String[] args){
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Your Full Name : ");
        String name = scanner.nextLine();
        name= name.toUpperCase();
        scanner.close();
        System.out.println("Initials: "+getInitials(name));
    }
    public static String getInitials(String name) {
        String[] words = name.split("\\s+");
        StringBuilder initials = new StringBuilder();
        for (int i = 0; i < words.length; i++) {
            initials.append(words[i].charAt(0));
            if (i < words.length - 1)
            {
                initials.append(".");
            }
        }
        return initials.toString();
    }
}
```

### AT TERMINAL:-

Enter Your Full Name :

Aashiq Rahaman

Initials: A.R

**PROBLEM: - 1**

1. Create a class Figure with method area and two member variables. Class Rectangle and class Triangle invoke super class constructor and compute their respective area. (using super keyword, method overriding, and runtime polymorphism).

**Solution:-**

```
class Figure {
    double side1;
    double side2;
    Figure(double side1, double side2) {
        this.side1 = side1;
        this.side2 = side2;
    }
    double computeArea() {
        System.out.println("Area computation is undefined for the generic
figure.");
        return 0;
    }
}
class Rectangle extends Figure {
    Rectangle(double length, double width) {
        super(length, width);
    }
    @Override
    double computeArea() {
        return side1 * side2;
    }
}
class Triangle extends Figure {
    Triangle(double base, double height) {
        super(base, height);
    }
    @Override
    double computeArea() {
        return 0.5 * side1 * side2;
    }
}
public class Main {
    public static void main(String[] args) {
        Figure rectangle = new Rectangle(10, 5);
        System.out.println("Area of Rectangle: " + rectangle.computeArea());
        Figure triangle = new Triangle(8, 4);
        System.out.println("Area of Triangle: " + triangle.computeArea());
        Figure genericFigure = new Figure(0, 0);
        genericFigure.computeArea();
    }
}
```

**AT TERMINAL:-**

Area of Rectangle: 50.0

Area of Triangle: 16.0

Area computation is undefined for the generic figure.

## PROBLEM: - 2

Create an abstract class 'Geometry' with an abstract method 'area' and a class member  $\text{PI}=3.14159$ . Circle and Rectangle are two subclasses of 'Geometry'. Each of the subclasses has area method to compute and display their area.

### Solution:-

```
abstract class Geometry {
    static final double PI = 3.14159;
    abstract void calc();
}
class Circle extends Geometry {
    private double r;
    Circle(double r) {
        this.r = r;
    }
    @Override
    void calc() {
        double res = PI * r * r;
        System.out.println("Area of Circle: " + res);
    }
}
class Rectangle extends Geometry {
    private double l;
    private double w;
    Rectangle(double l, double w) {
        this.l = l;
        this.w = w;
    }
    @Override
    void calc() {
        double res = l * w;
        System.out.println("Area of Rectangle: " + res);
    }
}
public class Main {
    public static void main(String[] args) {
        Geometry c = new Circle(5);
        c.calc();
        Geometry r = new Rectangle(10, 4);
        r.calc();
    }
}
```

### AT TERMINAL:-

Area of Circle: 78.53975

Area of Rectangle: 40.0

### PROBLEM: - 3

Create an interface **GeoAnalyzer** with member variable **PI=3.14159** and method **perimeter**. Class **circle** and **Rectangle** implement the interface **GeoAnalyzer**, compute perimeter and finally display it. Use runtime polymorphism.

### Solution:-

```
interface GeoAnalyzer {
    double PI = 3.14159;
    void peri();
}
class Circle implements GeoAnalyzer {
    private double rad;
    Circle(double rad) {
        this.rad = rad;    }
    @Override
    public void peri() {
        double res = 2 * PI * rad;
        System.out.println("Perimeter of Circle: " + res);
    }
}
class Rectangle implements GeoAnalyzer {
    private double len;
    private double wid;
    Rectangle(double len, double wid) {
        this.len = len;
        this.wid = wid;    }
    @Override
    public void peri() {
        double res = 2 * (len + wid);
        System.out.println("Perimeter of Rectangle: " + res);
    }
}
public class Main {
    public static void main(String[] args) {
        GeoAnalyzer c = new Circle(5);
        c.peri();
        GeoAnalyzer r = new Rectangle(10, 4);
        r.peri();
    }
}
```

### AT TERMINAL:-

Perimeter of Circle: 31.4159

Perimeter of Rectangle: 28.0

**PROBLEM: - 1**

**Write a java program to throw division by zero exception.**

**Solution:-**

```
package Jan16;

public class DivZero {
    public static void main(String[] args) {
        try {
            int numerator = 10;
            int denominator = 0;
            int result = numerator / denominator;
            System.out.println("The result is: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
            System.out.println("Exception message: " + e.getMessage());
        } finally {
            System.out.println("Execution of the try-catch block is complete.");
        }
    }
}
```

**AT TERMINAL:-**

```
$ javac DivZero.java
```

```
$ java DivZero
```

```
Error: Division by zero is not allowed.
Exception message: / by zero
Execution of the try-catch block is complete.
```

## PROBLEM: - 2

Create a thread using runnable interface.

**Solution:-**

```
package Jan16;
class MyRunnable implements Runnable {
    @Override
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Running in thread: " +
Thread.currentThread().getName() + " - Count: " + i);
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted: " +
Thread.currentThread().getName());
            }
        }
    }
}
public class Runnablethread {
    public static void main(String[] args) {
        MyRunnable myRunnable = new MyRunnable();
        Thread thread = new Thread(myRunnable);
        thread.start();
        for (int i = 1; i <= 5; i++) {
            System.out.println("Running in main thread: " +
Thread.currentThread().getName() + " - Count: " + i);
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println("Main thread interrupted.");
            }
        }
    }
}
```

**AT TERMINAL:-**

```
$ javac Runnablethread.java
```

```
$ java Runnablethread
```

```
Running in thread: Thread-0 - Count: 1
Running in main thread: main - Count: 1
Running in main thread: main - Count: 2
Running in thread: Thread-0 - Count: 2
Running in main thread: main - Count: 3
Running in thread: Thread-0 - Count: 3
Running in main thread: main - Count: 4
Running in thread: Thread-0 - Count: 4
Running in main thread: main - Count: 5
Running in thread: Thread-0 - Count: 5
```

oops