Manav Rachna International Institute of Research and Studies

School of Computer Applications

Programme	**Masters of Computer Applications**
Batch	**2025-27**
Semester	**1th Semester**
Roll No	**25/SCA/MCAN/072**
Class &Section	**MCA-1B**

OBJECT ORIENTED PROGRAMMING IN JAVA

(COURSE CODE : 6.0CA102C01H)

PRACTICAL FILE

Submitted To

Dr. Ritu Sachdeva

Submitted By

Abhishek Kumar Gupta

INDEX

S. No.	Aim of the Experiment	Signature	Grade
1	Write a program to find the factorial of n Number.		
2	Write a program to find the sequence of Fibonacci series up to n terms.		
3	Write a program to check whether given number is palindrome or not.		
4	Write a program to find HCF of two numbers.		
5	Write a Java Program that will display the sum of 1+1/2+1/3…..+1/n.		
6	Write a Java Program to find product of two matrices.		
7	Write a Java Program to find sum and subtraction of two matrices.		
8	Write a Java Program to sort the list in ascending Order.		
9	Write a Java Program to convert decimal into binary number.		
10	Write a Java Program to find largest and smallest of n numbers		
11	Write a Java program that shows the application of constructors.		
12	Write a Java program to find the electricity bill using inheritance		
13	Create a class “Vehicle” with instance variable vehicle_type. Inherit the class in a class called “Car” with instance model_type, company name etc. Display the information of the vehicle by defining the display() in both super and sub class.		

14	Create a class “Account” containing account No, and balance as an instance variable. Derive the Account class into two classes named “Savings” and “Current”. The “Savings” class should contain instance variable named interest Rate, and the “Current” class should contain instance variable called overdraft Limit. Define appropriate methods for all the classes to enable functionalities to check balance, deposit, and withdraw amount in Savings and Current account.		
15	Write a program to demonstrate the use of ‘this’ and ‘static’ keyword.		
16	Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, and Circle. Define one method area () in the abstract class and override this area () in these three subclasses to calculate for specific object i.e. area () of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.		
17	Write a java program to find the result sheet of a student using Interfaces.		
18	Write a java program which shows importing of classes from other packages.		
19	Assume that there are two packages, student and exam. A student package contains Student class and the exam package contains Result class. Write a program that generates mark sheet for students.		
20	Write a program to implement the concept of threading by implementing “Runnable” Interface.		
21	Write a program that executes two threads. One thread displays “Thread1” every 2,000 milliseconds, and the other displays “Thread2” every 4,000 milliseconds.		
22	Write a java program which use try and catch for exception handling.		
23	Write a java program which use multiple catch blocks.		
24	Write a java program which shows throwing our own exception.		

25	Write a program to handle Labels and Buttons using AWT Controls.		
26	Write a program to handle Check Boxes using AWT Controls		
27	Write a program to handle Lists and Scroll Bars using AWT Controls		

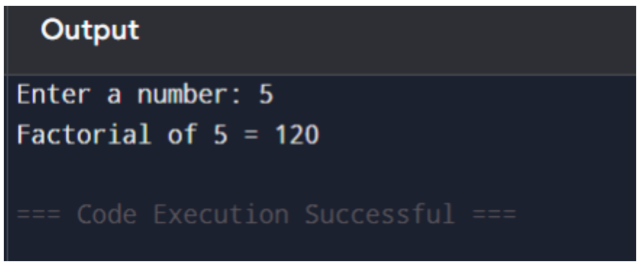
OBJECT ORIENTED PROGRAMMING IN JAVA

Abhishek Kr. Gupta | MCA 1 B | 25/SCA/MCAN/072

Question 1:
Write a program to find the factorial of n number.

```
import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter a number: ");
        int n = sc.nextInt();


        long fact = 1;


        for (int i = 1; i <= n; i++) {
            fact = fact * i;
        }


        System.out.println("Factorial of " + n + " = " + fact);


        sc.close();
```

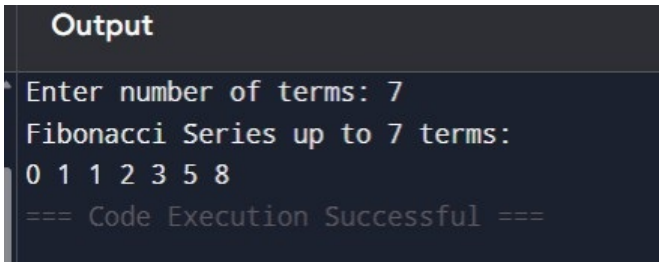
```
    }
}
```



Question 2:
Write a program to find the sequence of Fibonacci series up to n terms.

```
import java.util.Scanner;

public class FibonacciSeries {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of terms: ");
        int n = sc.nextInt();


        int a = 0, b = 1;

        System.out.println("Fibonacci Series up to " + n + " terms:");
        for (int i = 1; i <= n; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
```

```
        sc.close();
    }
}
```



Question 3:
Write a program to check whether given number is palindrome or not

```
import java.util.Scanner;

public class PalindromeNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        int original = n;
        int reverse = 0;

        while (n > 0) {
            int digit = n % 10;
            reverse = reverse * 10 + digit;
```

```
            n = n / 10;
        }

        if (original == reverse) {
            System.out.println(original + " is a Palindrome number.");
        } else {
            System.out.println(original + " is NOT a Palindrome number.");
        }

        sc.close();
    }
}
```

```
Output

Enter a number: 78
78 is NOT a Palindrome number.

=== Code Execution Successful ===
```

Question 4:
Write a program to find HCF of two numbers.

```
import java.util.Scanner;

public class HCFOfTwoNumbers {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

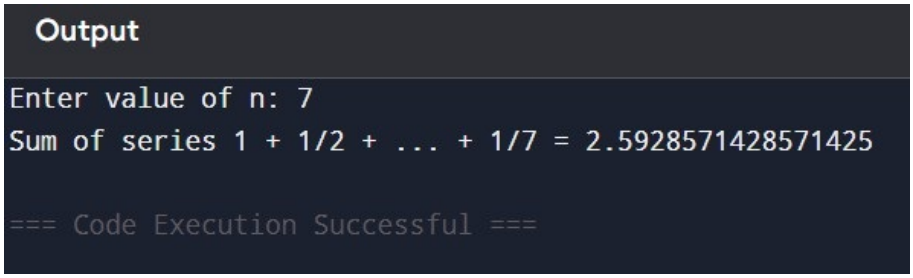
```
        System.out.print("Enter first number: ");
        int a = sc.nextInt();

        System.out.print("Enter second number: ");
        int b = sc.nextInt();

        int x = a;
        int y = b;

        // Euclid's algorithm
        while (y != 0) {
            int temp = y;
            y = x % y;
            x = temp;
        }

        System.out.println("HCF of " + a + " and " + b + " = " + x);

        sc.close();
    }
}
```

Output

```
Enter first number: 30
Enter second number: 60
HCF of 30 and 60 = 30

=== Code Execution Successful ===
```

Question 5:
Write a Java Program that will display the sum of 1 + 1/2 + 1/3 + … + 1/n.

```
import java.util.Scanner;

public class HarmonicSeriesSum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter value of n: ");
        int n = sc.nextInt();

        double sum = 0.0;

        for (int i = 1; i <= n; i++) {
            sum = sum + 1.0 / i;
        }

        System.out.println("Sum of series 1 + 1/2 + ... + 1/" + n + " = " + sum);

        sc.close();
    }
}
```



Question 6:
Write a Java Program to find product of two matrices.

```
import java.util.Scanner;

public class MatrixMultiplication {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // First matrix
        System.out.print("Enter rows of first matrix: ");
        int r1 = sc.nextInt();
        System.out.print("Enter columns of first matrix: ");
        int c1 = sc.nextInt();

        // Second matrix
        System.out.print("Enter rows of second matrix: ");
        int r2 = sc.nextInt();
        System.out.print("Enter columns of second matrix: ");
        int c2 = sc.nextInt();
```

```
        if (c1 != r2) {
            System.out.println("Matrix multiplication not possible (c1 must be equal to
r2).");
            sc.close();
            return;
        }

        int[][] A = new int[r1][c1];
        int[][] B = new int[r2][c2];
        int[][] C = new int[r1][c2];

        System.out.println("Enter elements of first matrix:");
        for (int i = 0; i < r1; i++) {
            for (int j = 0; j < c1; j++) {
                A[i][j] = sc.nextInt();
            }
        }

        System.out.println("Enter elements of second matrix:");
        for (int i = 0; i < r2; i++) {
            for (int j = 0; j < c2; j++) {
                B[i][j] = sc.nextInt();
            }
        }

        // Multiplication
        for (int i = 0; i < r1; i++) {
```

```
            for (int j = 0; j < c2; j++) {
                C[i][j] = 0;
                for (int k = 0; k < c1; k++) {
                    C[i][j] = C[i][j] + A[i][k] * B[k][j];
                }
            }
        }

        System.out.println("Product of the two matrices:");
        for (int i = 0; i < r1; i++) {
            for (int j = 0; j < c2; j++) {
                System.out.print(C[i][j] + " ");
            }
            System.out.println();
        }

        sc.close();
    }
}
```

Output

```
Enter rows of first matrix: 2
Enter columns of first matrix: 2
Enter rows of second matrix: 2
Enter columns of second matrix: 2
Enter elements of first matrix:
1 2
3 4
Enter elements of second matrix:
5 6
7 8
Product of the two matrices:
19 22
43 50

=== Code Execution Successful ===
```

Question 7:
Write a Java Program to find sum and subtraction of two matrices.

```
import java.util.Scanner;


public class MatrixSumSubtraction {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.print("Enter number of rows: ");
        int r = sc.nextInt();


        System.out.print("Enter number of columns: ");
        int c = sc.nextInt();


        int[][] A = new int[r][c];
        int[][] B = new int[r][c];
        int[][] sum = new int[r][c];
        int[][] diff = new int[r][c];


        System.out.println("Enter elements of first matrix:");
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                A[i][j] = sc.nextInt();
            }
        }
```

```
System.out.println("Enter elements of second matrix:");
for (int i = 0; i < r; i++) {
    for (int j = 0; j < c; j++) {
        B[i][j] = sc.nextInt();
    }
}

// Sum and Subtraction
for (int i = 0; i < r; i++) {
    for (int j = 0; j < c; j++) {
        sum[i][j] = A[i][j] + B[i][j];
        diff[i][j] = A[i][j] - B[i][j];
    }
}

System.out.println("Sum of two matrices:");
for (int i = 0; i < r; i++) {
    for (int j = 0; j < c; j++) {
        System.out.print(sum[i][j] + " ");
    }
    System.out.println();
}

System.out.println("Subtraction of two matrices (A - B):");
for (int i = 0; i < r; i++) {
    for (int j = 0; j < c; j++) {
```

```
                System.out.print(diff[i][j] + " ");
            }
            System.out.println();
        }

        sc.close();
    }
}
```

Output

```
Enter number of rows: 2
Enter number of columns: 2
Enter elements of first matrix:
1 2
3 4
Enter elements of second matrix:
4 3
2 1
Sum of two matrices:
5 5
5 5
Subtraction of two matrices (A - B):
-3 -1
1 3
```

Question 8:
Write a Java Program to sort the list in ascending order.

```
import java.util.Scanner;

public class SortAscending {
    public static void main(String[] args) {
```

```
Scanner sc = new Scanner(System.in);

System.out.print("Enter number of elements: ");
int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter " + n + " numbers:");
for (int i = 0; i < n; i++) {
    arr[i] = sc.nextInt();
}

// Bubble sort
for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - 1 - i; j++) {
        if (arr[j] > arr[j + 1]) {
            int temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
}

System.out.println("Sorted list in ascending order:");
for (int i = 0; i < n; i++) {
    System.out.print(arr[i] + " ");
```

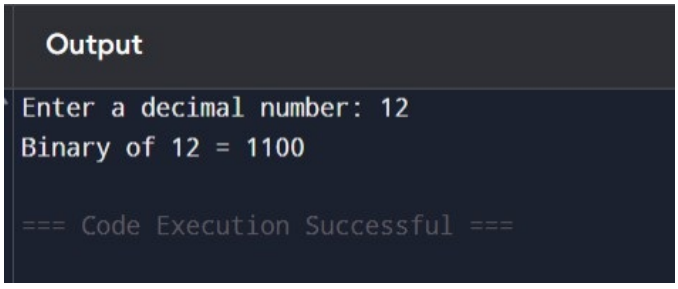
```
        }

        sc.close();
    }
}
```

```
Output
Enter number of elements: 5
Enter 5 numbers:
1 12 5 3 4
Sorted list in ascending order:
1 3 4 5 12
=== Code Execution Successful ===
```

Question 9:
Write a Java Program to convert decimal into binary number.

```
import java.util.Scanner;

public class DecimalToBinary {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a decimal number: ");
        int n = sc.nextInt();

        if (n == 0) {
            System.out.println("Binary: 0");
            sc.close();
```

```
            return;
        }

        String binary = "";

        int num = n;
        while (num > 0) {
            int rem = num % 2;
            binary = rem + binary; // add remainder at beginning
            num = num / 2;
        }

        System.out.println("Binary of " + n + " = " + binary);

        sc.close();
    }
}
```



Question 10:
Write a Java Program to find largest and smallest of n numbers.

```
import java.util.Scanner;
```

```
public class LargestSmallest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter " + n + " numbers:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int smallest = arr[0];
        int largest = arr[0];

        for (int i = 1; i < n; i++) {
            if (arr[i] < smallest) {
                smallest = arr[i];
            }
            if (arr[i] > largest) {
                largest = arr[i];
            }
        }
```

```
        System.out.println("Smallest number = " + smallest);
        System.out.println("Largest number  = " + largest);

        sc.close();
    }
}
```

```
Output
Enter number of elements: 5
Enter 5 numbers:
34 56 23 12 2
Smallest number = 2
Largest number  = 56
```

11. Question:

Write a Java program that shows the application of constructors.

```
public class ConstructorDemo {
    public static void main(String[] args) {
        Student s1 = new Student("Aastha", 21);
        s1.display();
    }
}

class Student {
    String name;
```

```
    int age;

    // Constructor
    Student(String n, int a) {
        name = n;
        age = a;
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
```

Q12. Write a Java program to find the electricity bill using inheritance.

```
public class Main {
    public static void main(String[] args) {
        Bill b1 = new Bill("Tisha", 250);   // 250 units ka bill
        b1.display();

        System.out.println();
```

```
        Bill b2 = new Bill("Aastha", 90);   // 90 units ka bill
        b2.display();
    }
}

class Customer {
    String name;
    int units;

    Customer(String name, int units) {
        this.name = name;
        this.units = units;
    }
}

class Bill extends Customer {

    Bill(String name, int units) {
        super(name, units);
    }

    double calculateBill() {
        // Simple rate logic:
        // 0–100 units  : ₹5 per unit
        // 101–200 units: ₹7 per unit
```

```
        // >200 units   : ₹10 per unit
        if (units <= 100) {
            return units * 5;
        } else if (units <= 200) {
            return units * 7;
        } else {
            return units * 10;
        }
    }

    void display() {
        System.out.println("Customer Name: " + name);
        System.out.println("Units Used   : " + units);
        System.out.println("Total Bill   : ₹" + calculateBill());
    }
}
```

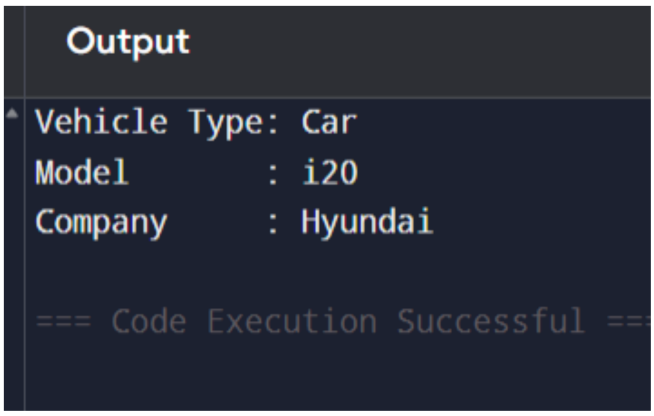
```
Output
Customer Name: Tisha
Units Used   : 250
Total Bill   : 2500.0

Customer Name: Aastha
Units Used   : 90
Total Bill   : 450.0

=== Code Execution Successful ===
```

Q13. Create a class “Vehicle” with instance variable vehicle_type. Inherit the class in a class called “Car” with instance model_type, company name etc.

Display the information of the vehicle by defining the display() in both super and sub class.

```
public class Main {
    public static void main(String[] args) {
        Car c = new Car("Car", "i20", "Hyundai");
        c.display();
    }
}

class Vehicle {
    String vehicle_type;

    Vehicle(String type) {
        this.vehicle_type = type;
    }

    void display() {
        System.out.println("Vehicle Type: " + vehicle_type);
    }
}

class Car extends Vehicle {
    String model;
    String company;
```

```
    Car(String type, String model, String company) {
        super(type);
        this.model = model;
        this.company = company;
    }

    // Overriding display()
    void display() {
        super.display();  // Vehicle ka display()
        System.out.println("Model          : " + model);
        System.out.println("Company        : " + company);
    }
}
```



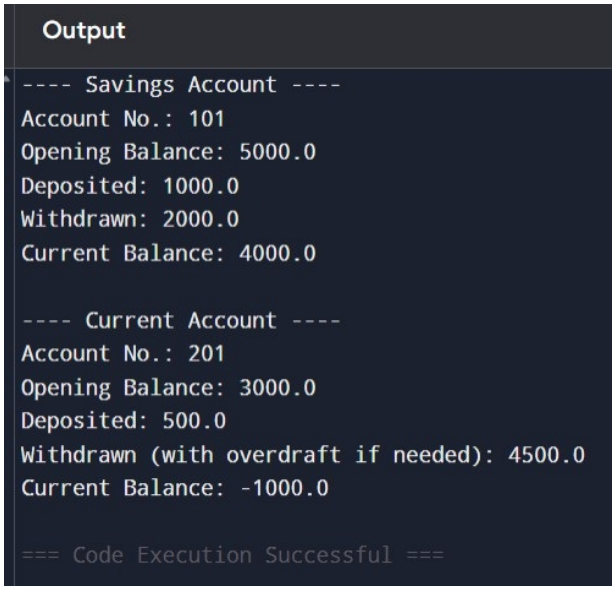
Q14. Create a class “Account” containing account No, and balance as an instance variable. Derive the Account class into two classes named “Savings” and “Current”. The “Savings” class should contain instance variable named interest Rate, and the “Current” class should contain instance variable called overdraft Limit. Define appropriate methods for all the classes to enable functionalities to check balance, deposit, and withdraw amount in Savings and Current account.

```
public class Main {
    public static void main(String[] args) {

        Savings s = new Savings(101, 5000.0, 5.5);
        System.out.println("---- Savings Account ----");
        s.showAccountInfo();
        s.deposit(1000);
        s.withdraw(2000);
        s.showBalance();

        System.out.println();

        Current c = new Current(201, 3000.0, 2000.0);
        System.out.println("---- Current Account ----");
        c.showAccountInfo();
        c.deposit(500);
        c.withdraw(4500);   // overdraft use karega
        c.showBalance();
    }
}

class Account {
    int accNo;
    double balance;

    Account(int accNo, double balance) {
```

```
    this.accNo = accNo;
    this.balance = balance;
}

void deposit(double amount) {
    balance += amount;
    System.out.println("Deposited: " + amount);
}

void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
    } else {
        System.out.println("Insuficient balance!");
    }
}

void showBalance() {
    System.out.println("Current Balance: " + balance);
}

void showAccountInfo() {
    System.out.println("Account No.: " + accNo);
    System.out.println("Opening Balance: " + balance);
}
```

```
}

class Savings extends Account {
    double interestRate;

    Savings(int accNo, double balance, double interestRate) {
        super(accNo, balance);
        this.interestRate = interestRate;
    }
}

class Current extends Account {
    double overdraftLimit;

    Current(int accNo, double balance, double overdraftLimit) {
        super(accNo, balance);
        this.overdraftLimit = overdraftLimit;
    }

    // Overriding withdraw for overdraft facility
    void withdraw(double amount) {
        if (balance + overdraftLimit >= amount) {
            balance -= amount;
            System.out.println("Withdrawn (with overdraft if needed): " + amount);
        } else {
            System.out.println("Amount exceeds overdraft limit!");
```

```
    }
  }
}
```



Q15. Write a program to demonstrate the use of ‘this’ and ‘static’ keyword.

```
public class Main {
   public static void main(String[] args) {

       Demo d1 = new Demo("Aastha");
       Demo d2 = new Demo("Tisha");
       Demo d3 = new Demo("Riya");

       d1.display();
       d2.display();
       d3.display();
```

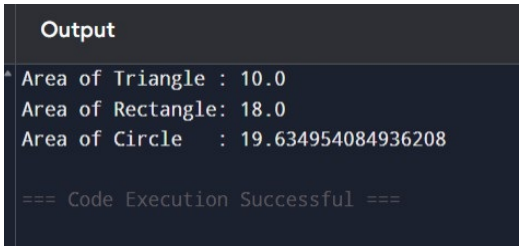
```
        // static variable class name se access hota hai
        System.out.println("Total Objects Created: " + Demo.count);
    }
}

class Demo {
    static int count = 0;   // static variable (shared by all objects)
    String name;

    Demo(String name) {
        this.name = name;   // 'this' current object ko refer karta hai
        count++;            // har object banne par count++
    }

    void display() {
        System.out.println("Object Name: " + this.name);
    }
}
```

Q16. Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, and Circle. Define one method area () in the abstract class and override this area () in these three subclasses to calculate for specific object i.e. area () of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.

```
abstract class Shape {
    abstract double area();         // abstract method
}

class Triangle extends Shape {
    double base, height;

    Triangle(double b, double h) {
        this.base = b;
        this.height = h;
    }

    @Override
    double area() {
        return 0.5 * base * height;
    }
}

class Rectangle extends Shape {
    double length, breadth;
```

```
    Rectangle(double l, double b) {
        this.length = l;
        this.breadth = b;
    }

    @Override
    double area() {
        return length * breadth;
    }
}

class Circle extends Shape {
    double radius;

    Circle(double r) {
        this.radius = r;
    }

    @Override
    double area() {
        return Math.PI * radius * radius;
    }
}

public class Main {
    public static void main(String[] args) {
```

```
        Triangle t = new Triangle(5, 4);
        Rectangle r = new Rectangle(6, 3);
        Circle c = new Circle(2.5);

        System.out.println("Area of Triangle : " + t.area());
        System.out.println("Area of Rectangle: " + r.area());
        System.out.println("Area of Circle   : " + c.area());
    }
}
```

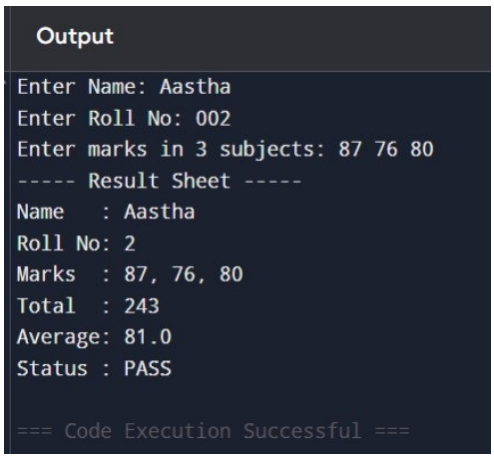


Q17. Write a Java program to find the result sheet of a student using Interfaces

```
import java.util.Scanner;

interface Result {
    void calculateResult();
}

class Student implements Result {
    String name;
    int rollNo;
    int m1, m2, m3;
```

```
    Student(String name, int rollNo, int m1, int m2, int m3) {
        this.name = name;
        this.rollNo = rollNo;
        this.m1 = m1;
        this.m2 = m2;
        this.m3 = m3;
    }

    @Override
    public void calculateResult() {
        int total = m1 + m2 + m3;
        double average = total / 3.0;
        String status = (average >= 40) ? "PASS" : "FAIL";

        System.out.println("----- Result Sheet -----");
        System.out.println("Name   : " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Marks  : " + m1 + ", " + m2 + ", " + m3);
        System.out.println("Total  : " + total);
        System.out.println("Average: " + average);
        System.out.println("Status : " + status);
    }
}

public class Main {
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Name: ");
        String name = sc.nextLine();

        System.out.print("Enter Roll No: ");
        int roll = sc.nextInt();

        System.out.print("Enter marks in 3 subjects: ");
        int m1 = sc.nextInt();
        int m2 = sc.nextInt();
        int m3 = sc.nextInt();

        Student s = new Student(name, roll, m1, m2, m3);
        s.calculateResult();

        sc.close();
    }
}
```



Q18. Write a Java program which shows importing of classes from other packages.

```
class MyMessage {
    public void show() {
        System.out.println("Hello from another package class (conceptually)!");
    }
}

public class Main {
    public static void main(String[] args) {

        MyMessage msg = new MyMessage();
        msg.show();
    }
}
```

```
Output
Hello from another package class (conceptually)!

=== Code Execution Successful ===
```

Q19. Assume that there are two packages, student and exam. A student package contains Student class and the exam package contains Result class. Write a program that generates mark sheet for students.

```
class StudentInfo {
    String name;
    int rollNo;
```

```
    StudentInfo(String name, int rollNo) {
        this.name = name;
        this.rollNo = rollNo;
    }
}

class ResultSheet {
    StudentInfo student;
    int m1, m2, m3;

    ResultSheet(StudentInfo s, int m1, int m2, int m3) {
        this.student = s;
        this.m1 = m1;
        this.m2 = m2;
        this.m3 = m3;
    }

    void printMarkSheet() {
        int total = m1 + m2 + m3;
        double per = total / 3.0;

        System.out.println("------ Mark Sheet ------");
        System.out.println("Name   : " + student.name);
        System.out.println("Roll No: " + student.rollNo);
        System.out.println("Marks  : " + m1 + ", " + m2 + ", " + m3);
```

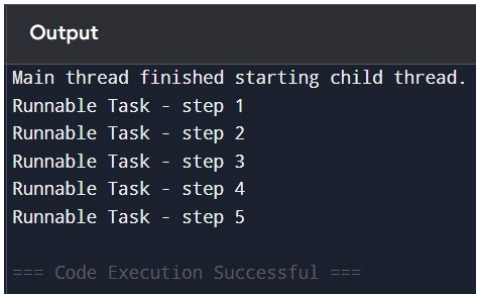
```
        System.out.println("Total  : " + total);
        System.out.println("Percent: " + per + "%");
    }
}

public class Main {
    public static void main(String[] args) {
        StudentInfo s = new StudentInfo("Tisha", 101);
        ResultSheet r = new ResultSheet(s, 85, 78, 92);
        r.printMarkSheet();
    }
}
```

```
Output
------ Mark Sheet ------
Name   : Tisha
Roll No: 101
Marks  : 85, 78, 92
Total  : 255
Percent: 85.0%

=== Code Execution Successful ===
```

Q20. Write a program to implement the concept of threading by implementing “Runnable” Interface.

```
class MyTask implements Runnable {
    String taskName;

    MyTask(String name) {
        this.taskName = name;
    }
```

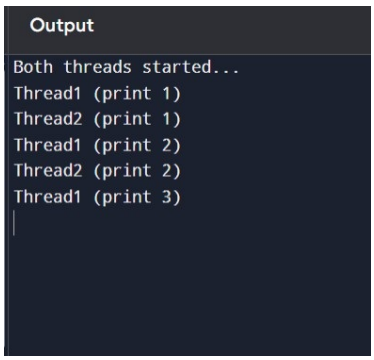
```
    @Override
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(taskName + " - step " + i);
            try {
                Thread.sleep(500);   // 0.5 second
            } catch (InterruptedException e) {
                System.out.println("Interrupted");
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        MyTask t1 = new MyTask("Runnable Task");

        Thread th = new Thread(t1); // Runnable ko Thread me wrap kiya
        th.start();

        System.out.println("Main thread finished starting child thread.");
    }
}
```



Q21. Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds.

```
class MessageTask implements Runnable {
    String message;
    int delay;   // in milliseconds

    MessageTask(String message, int delay) {
        this.message = message;
        this.delay = delay;
    }

    @Override
    public void run() {

        for (int i = 1; i <= 5; i++) {
            System.out.println(message + " (print " + i + ")");
            try {
                Thread.sleep(delay);
            } catch (InterruptedException e) {
```

```
            System.out.println(message + " interrupted");
        }
    }
  }
}

public class Main {
  public static void main(String[] args) {
    MessageTask task1 = new MessageTask("Thread1", 2000); // 2 seconds
    MessageTask task2 = new MessageTask("Thread2", 4000); // 4 seconds

    Thread t1 = new Thread(task1);
    Thread t2 = new Thread(task2);

    t1.start();
    t2.start();

    System.out.println("Both threads started...");
  }
}
```



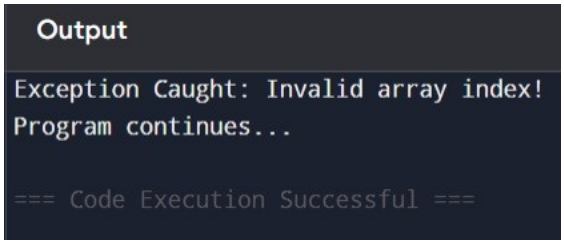
Q22. Write a java program which use try and catch for exception handling.

```
public class Main {
    public static void main(String[] args) {

        try {
            int a = 10;
            int b = 0;
            System.out.println("Result = " + (a / b));  // error
        }
        catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }
        finally {
            System.out.println("Finally block always executes.");
        }

        System.out.println("Program continues...");
    }
}
```

```
Output
ERROR!
Error: Division by zero is not allowed.
Finally block always executes.
Program continues...

=== Code Execution Successful ===
```

Q23. Write a java program which use multiple catch blocks.

```
public class Main {
    public static void main(String[] args) {

        try {
            int[] arr = {10, 20, 30};
            System.out.println(arr[5]);

            int a = 10 / 0;
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception Caught: Invalid array index!");
        }
        catch (ArithmeticException e) {
            System.out.println("Exception Caught: Division by zero!");
        }
        catch (Exception e) {
            System.out.println("Exception Caught: Some other exception
occurred!");
        }

        System.out.println("Program continues...");
    }
}
```



Q24. Write a java program which shows throwing our own exception.

```
class MyException extends Exception {
    MyException(String message) {
        super(message);
    }
}

public class Main {
    public static void main(String[] args) {

        int age = 15;

        try {

            if (age < 18) {
                throw new MyException("Age must be 18 or above!");
            }

            System.out.println("Valid age.");
        }
```

```
        catch (MyException e) {
            System.out.println("Custom Exception Caught: " + e.getMessage());
        }
    }
}
```

```
Output
Custom Exception Caught: Age must be 18 or above!

=== Code Execution Successful ===
```

Q25. Write a program to handle Labels and Buttons using AWT Controls.

```
import java.awt.*;
import java.awt.event.*;

class MyFrame extends Frame implements ActionListener {

    Label label;
    Button btnOk, btnCancel;

    MyFrame() {
        // Frame title
        setTitle("AWT Label and Button Example");

        // Layout set
        setLayout(new FlowLayout());
```

```
// Label create
label = new Label("Click a button...");

// Buttons create
btnOk = new Button("OK");
btnCancel = new Button("Cancel");

// Buttons ko action listener attach
btnOk.addActionListener(this);
btnCancel.addActionListener(this);

// Components ko frame me add
add(label);
add(btnOk);
add(btnCancel);

// Frame size & visibility
setSize(300, 150);
setVisible(true);

// Window close - listener
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        dispose();          // window band karo
    }
});
```

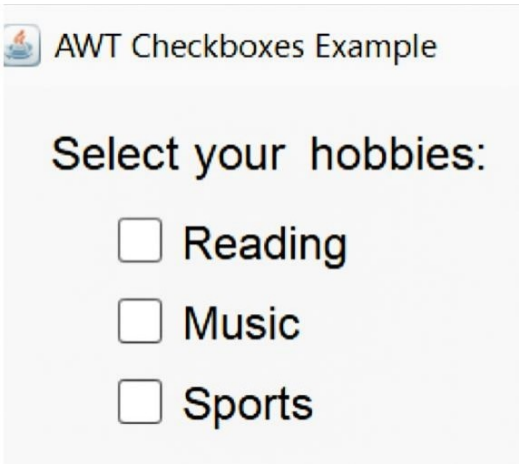
```
    }

    // Button click handle
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btnOk) {
            label.setText("OK Button Clicked");
        } else if (e.getSource() == btnCancel) {
            label.setText("Cancel Button Clicked");
        }
    }
}

public class Main {
    public static void main(String[] args) {
        new MyFrame();   // Frame create karo
    }
}
```

Q.26 Write a program to handle Check Boxes using AWT Controls

```
import java.awt.*;
```

```
import java.awt.event.*;

class MyFrame extends Frame implements ItemListener {

    Label l;
    Checkbox c1, c2, c3;

    MyFrame() {
        setLayout(new FlowLayout());

        l = new Label("Select your hobbies:");

        c1 = new Checkbox("Reading");
        c2 = new Checkbox("Music");
        c3 = new Checkbox("Sports");

        c1.addItemListener(this);
        c2.addItemListener(this);
        c3.addItemListener(this);

        add(l);
        add(c1);
        add(c2);
        add(c3);

        setSize(300, 200);
```

```
        setVisible(true);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    public void itemStateChanged(ItemEvent e) {
        String msg = "Selected: ";

        if (c1.getState()) msg += "Reading ";
        if (c2.getState()) msg += "Music ";
        if (c3.getState()) msg += "Sports ";

        l.setText(msg);
    }
}

public class Main {
    public static void main(String[] args) {
        new MyFrame();
    }
}
```



Q.27 Write a program to handle Lists and Scroll Bars using AWT Controls

```
import java.awt.*;
import java.awt.event.*;

class MyFrame extends Frame implements ItemListener, AdjustmentListener {

    List list;
    Scrollbar sb;
    Label l1, l2;

    MyFrame() {
        setLayout(new FlowLayout());

        l1 = new Label("Selected item: ");
        l2 = new Label("Scroll value: 0");

        list = new List(4); // visible rows
```

```
    list.add("Red");
    list.add("Green");
    list.add("Blue");
    list.add("Yellow");
    list.add("Black");
    list.add("White");

    sb = new Scrollbar(Scrollbar.HORIZONTAL, 0, 10, 0, 110);

    list.addItemListener(this);
    sb.addAdjustmentListener(this);

    add(list);
    add(sb);
    add(l1);
    add(l2);

    setSize(350, 200);
    setVisible(true);

    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            dispose();
        }
    });
}
```

```
    public void itemStateChanged(ItemEvent e) {
        String item = list.getSelectedItem();
        l1.setText("Selected item: " + item);
    }

    public void adjustmentValueChanged(AdjustmentEvent e) {
        int value = sb.getValue();
        l2.setText("Scroll value: " + value);
    }
}

public class Main {
    public static void main(String[] args) {
        new MyFrame();
    }
}
```