

# College of Engineering, Thiruvananthapuram

## Object Oriented Programming Lab



Anirudh A. V.

S2 CSE R2, Roll No. 11

Department of Computer Science  
& Engineering

January 24, 2022

# 1 Pascal Triangle

## 1.1 Aim

To write a Java program to print the pattern shown below:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

## 1.2 Algorithm

Step - 1 Start

Step - 2 Declare variables i=0, j, and n. initialise n = 4.

Step - 3 Create a static method called factorial(num) which returns the factorial of the input parameter.

Step - 4 Create a static method called nCr(n,r) which returns the combination of two numbers n & r which are passed as input parameters.

Step - 5 Repeat the following until i<=n

Step - 5.1. For j=0 repeat the following until j<=n-i

Step - 5.1.1. Print “ ”.

Step - 5.1.2. Increment j.

Step - 5.2. For j=0 repeat the following until j<=n

Step - 5.2.1. Print “ nCr(i,j)”

Step - 5.2.2. Increment j

Step - 5.3. Increment i

Step - 6 Stop

## 1.3 Code

```
/** This program is intended to print the pascals Triangle.
    This is written by Anirudh A V on 24 Jan 2022*/

public class PascalsTriangle {
    static int factorial(int n) {

        int fact;

        for(fact = 1; n > 1; n--){
            fact *= n;
        }
        return fact;
    }
    static int nCr(int n,int r) {
        return factorial(n) / ( factorial(n-r) * factorial(r) );
    }
    public static void main(String args[]){

        System.out.println("");

        int n, i, j;
        n = 4;

        for(i = 0; i <= n; i++) {
            for(j = 0; j <= n-i; j++){
                System.out.print(" ");
            }
            for(j = 0; j <= i; j++){
                System.out.print(" "+nCr(i, j));
            }
            System.out.println();
        }
    }
}
```

## 1.4 Sample Output

```
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_1\Lab_exam> javac PascalsTriangle.java
PS E:\Anirudh\Anirudh\CET\SEM 3\OOP_Lab\Java_cycle_1\Lab_exam> java PascalsTriangle
```

```
  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

## 2 Diagonal sum

### 2.1 Aim

To write a Java program to find the sum of diagonal elements of a given matrix

### 2.2 Algorithm

Step - 1 Start

Step - 2 Declare variables row, col, and a 2D array matrix, Initialize sum = 0.

Step - 3 Take in input from the user for the order of the matrix and store it in row and col.

Step - 4 Take the input for the matrix and store it in 'matrix'.

Step - 5 For i=0 repeat until i<row

Step - 5.1. For j=0 repeat until j<col

Step - 5.1.1. If i == j then,

Step - 5.1.1.1. sum = sum + matrix[i][j]

Step - 5.1.2. Increment j

Step - 5.2. Increment i

Step - 6 Print sum as the sum of diagonal elements.

Step - 7 Stop

## 2.3 Code

```
import java.util.Scanner;

public class diagonal_sum {

    public static void print_matrix(int[][] matrix, int row, int col) {
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                System.out.printf(matrix[i][j] + " ");
            }
            System.out.printf("\n");
        }
    }

    public static int sum(int[][] matrix, int row, int col) {
        int sum = 0;
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                if (i == j) {
                    sum = sum + matrix[i][j];
                }
            }
        }
        return sum;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.printf("\nEnter the order of the matrix (mxn) : ");
        int row = sc.nextInt();
        int col = sc.nextInt();

        int[][] matrix = new int[row][col];

        System.out.println("Enter the matrix : ");
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        sc.close();

        System.out.println("\nMatrix : ");
        print_matrix(matrix, row, col);
        int sum = sum(matrix, row, col);
        System.out.printf("Sum of diagonal elements : "+sum );
    }
}
```

## 2.4 Sample Output

Enter the order of the matrix (mxn) : 3 3

Enter the matrix :

1 2 3 4 5 6 7 8 9

Matrix :

1 2 3

4 5 6

7 8 9

Sum of diagonal elements : 15

PS E:\Anirudh\Anirudh\CET\SEM 3\OOP\_Lab\Java\_cycle\_1\Lab\_exam> █

---

## 3 Smallest and Largest Element

### 3.1 Aim

To write a Java program to find the smallest and largest element in an integer array.

### 3.2 Algorithm

- Step - 1 Start
- Step - 2 Declare a variable n and an integer array 'array'.
- Step - 3 Initialize n as the number of elements of the array from the user input
- Step - 4 Initialize an Integer array and store the input from the user in it.
- Step - 5 Copy the elements of the array to another array by creating a function.
- Step - 6 Sort the copied array using bubble sort ascending order.
- Step - 7 Display array[0] as the smallest element and array[n-1] as the largest.
- Step - 8 Stop



### 3.3 Code

```
/**
 * This program is intended to determine the smallest and the largest
 * element of an integer array.
 * This is written by Anirudh A V on 24 Jan 2022
 */

import java.util.Scanner;

public class Integer_array {
    public static void smallest_largest(int[] array, int n) {
        int temp;
        for(int i = 0; i < n; i++){
            for(int j = i + 1; j < n; j++){
                if (array[i] > array[j]) {
                    temp = array[i];
                    array[i] = array[j];
                    array[j] = temp;
                }
            }
        }
        System.out.println("The smallest number is "+array[0]);
        System.out.println("The largest number is "+array[n-1]);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.printf("\nEnter the number of elements in the array : ");
        int n = sc.nextInt();
        int[] array = new int[30];
        System.out.println("Enter the array elements : ");
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        sc.close();
        System.out.println("\nThe elements of array : ");
        for (int i = 0; i < n; i++) {
            System.out.println(array[i] + " ");
        }
        smallest_largest(array, n);
    }
}
```

### 3.4 Sample Output

Enter the number of elements in the array : 5

Enter the array elements :

12 23 43 76 8

The elements of array :

12

23

43

76

8

The smallest number is 8

The largest number is 76

PS E:\Anirudh\Anirudh\CET\SEM 3\OOP\_Lab\Java\_cycle\_1\Lab\_exam> █