**4. Develop a C# program to demonstrate Division by Zero and Index Out of Range exceptions.**

**Program :-**

using System;

class Program

{

static void Main()

{

// Division by Zero Exception

DivideByZeroExceptionExample();

// Index Out of Range Exception

IndexOutOfRangeExceptionExample();

Console.ReadLine(); // Keep the console window open

}

static void DivideByZeroExceptionExample()

{

try

{

int numerator = 10;

int denominator = 0;

int result = numerator / denominator; // Division by zero will throw an exception

Console.WriteLine($"Result of division: {result}");

}

catch (DivideByZeroException ex)

{

Console.WriteLine($"Error: {ex.Message}");

}

}

static void IndexOutOfRangeExceptionExample()

{

try

{

int[] numbers = { 1, 2, 3, 4, 5 };

int index = 10; // Accessing an index that is out of range will throw an exception

int value = numbers[index];

Console.WriteLine($"Value at index {index}: {value}");

}

catch (IndexOutOfRangeException ex)

{

Console.WriteLine($"Error: {ex.Message}");

}

}

}

Output :-

ERROR!

Error: Attempted to divide by zero.

ERROR!

Error: Index was outside the bounds of the array.

**5. Develop a C# program to generate and printPascal Triangle using Two Dimensional arrays.**

**Program :-**

using System;

namespace PascalTriangleDemo {

class Example {

public static void Main() {

int rows = 5, val = 1, blank, i, j;

Console.WriteLine("Pascal's triangle");

for(i = 0; i<rows; i++) {

for(blank = 1; blank <= rows-i; blank++)

Console.Write(" ");

for(j = 0; j <= i; j++) {

if (j == 0||i == 0)

val = 1;

else

val = val\*(i-j+1)/j;

Console.Write(val + " ");

}

Console.WriteLine();

}

}

}

}

Output :-

Pascal's triangle

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

**6. Develop a C# program to generate and print Floyds Triangle using Jagged arrays.**

**using System;**

**Program:-**

class Program

{

static void Main()

{

Console.Write("Enter the number of rows for Floyd's Triangle: ");

if (int.TryParse(Console.ReadLine(), out int rows))

{

int[][] floydsTriangle = GenerateFloydsTriangle(rows);

Console.WriteLine("Floyd's Triangle:");

PrintFloydsTriangle(floydsTriangle);

}

else

{

Console.WriteLine("Invalid input. Please enter a valid number of rows.");

}

}

static int[][] GenerateFloydsTriangle(int rows)

{

int[][] triangle = new int[rows][];

int count = 1;

for (int i = 0; i < rows; i++)

{

triangle[i] = new int[i + 1];

for (int j = 0; j <= i; j++)

{

triangle[i][j] = count++;

}

}

return triangle;

}

static void PrintFloydsTriangle(int[][] triangle)

{

foreach (int[] row in triangle)

{

foreach (int number in row)

{

Console.Write(number + " ");

}

Console.WriteLine();

}

}

}

Output :-

Enter the number of rows for Floyd's Triangle: 10

Floyd's Triangle:

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31 32 33 34 35 36

37 38 39 40 41 42 43 44 45

46 47 48 49 50 51 52 53 54 55