

DEV SANSKRITI VISHWAVIDYALAYA



SESSION 2018-21

Practical file

On

“C#.NET”

Submitted to:

Mr. Chandrashekhar Patel

Dept. of computer science

Submitted by:

Rupali Singh

B.C.A 5th Sem

INDEX

S. No.	Task	Remark
1	Write a program to print an Armstrong Number	
2	Write a program to print factorial of a number	
3	Write a program to find the GCD of two numbers	
4	Write a program to check if a number is prime number	
5	Write a program to print the Fibonacci series	
6	Write a program to print the half pyramid pattern	
7	Write a program to print the half pyramid pattern with numbers	
8	Write a program to print the half pyramid inverse pattern	
9	Write a program to print the pyramid pattern	
10	Write a program to print the inverse pyramid pattern	
11	Write a program to print the diamond pattern	
12	Write a program to print pascal's triangle	
13	Write a program to compare two string without using string library functions	

14	Write a program to count a total number of alphabets, digits and special characters in a string	
15	Write a program to copy one string into another string	
16	Write a program to find maximum occurring character in a string	
17	Write a program to check whether a given substring is present in the given string	
18	Write a program for Abstraction	
19	Write a program for single inheritance	
20	Write a program for multilevel inheritance	
21	Write a program for multiple inheritance	
22	Write a program for method overloading	
23	Write a program for method overriding	
24	Write a program for interface	
25	Write a program for exception handling through try and catch	
26	Write a program for properties	
27	Write a program for threading	
28	Write a program to access data from database using ADO.NET	
29	Write a program for namespace	
30	Write a program to print square pattern	

Task-01: Write a program to print an Armstrong Number.

Program:

```
References
static void Main(string[] args)
{
    int num, rem, arm=0 ,given_number;
    Console.WriteLine("Enter the number: ");
    num = int.Parse(Console.ReadLine());
    given_number = num;
    while (num > 10)
    {
        rem = num % 10;
        num /= 10;
        arm += rem * rem * rem;
    }
    arm += num * num * num;
    if (arm == given_number)
    {
        Console.WriteLine("It's an Armstrong Number");
    }
    else
    {
        Console.WriteLine("Not an Armstrong Number");
    }
}
```

Output:

```
Enter the number:
153
It's an Armstrong Number

C:\Users\MY\source\repos\File\File\bin\Debug\netcoreapp
To automatically close the console when debugging stops
```

```
Enter the number:
121
Not an Armstrong Number

C:\Users\MY\source\repos\File\File\bin\Debug\netco
To automatically close the console when debugging :
le when debugging stops
```

Task-02: Write a program to print factorial of a number

Program:

```
using System;

namespace File
{
    References
    class Program
    {
        References
        static void Main(string[] args)
        {
            int i, n, fact = 1;
            Console.WriteLine("Enter the number: ");
            n = int.Parse(Console.ReadLine());
            for (i = n; i >= 1; i--)
            {
                fact *= i;
            }
            Console.WriteLine("Factorial of " + n + " is : " + fact);
        }
    }
}
```

Output:

```
Microsoft Visual Studio Debug Console

Enter the number:
6
Factorial of 6 is : 720

C:\Users\MY\source\repos\File\File\bin\Debug\netcoreapp3.1\File.exe (process)
To automatically close the console when debugging stops, enable Tools->Options->Debug Console->Close console when debugging stops.
Press any key to close this window . . .
```

Task-03: Write a program to find the GCD of two numbers

Program:

```

static void Main(string[] args)
{
    Console.WriteLine("Enter two numbers: ");
    int a = int.Parse(Console.ReadLine());
    int b = int.Parse(Console.ReadLine());
    int temp, remainder;
    if (b > a)
    {
        temp = b;
        b = a;
        a = temp;
    }
    while (a != 0 && b != 0)
    {
        remainder = a % b;
        a = b;
        b = remainder;
    }
    if (a == 0 && b != 0)
    {
        Console.WriteLine("GCD: " + b);
    }
    else if (a != 0 && b == 0)
    {
        Console.WriteLine("GCD: " + a);
    }
}
}

```

Output:

```

Enter two numbers:
192
270
GCD: 6
C:\Users\MY\source\repos\File\File\bin\Deb

```

Task-04: Write a program to check if number is a prime number

Program:

```
number1 | primenumber1 | Main
using System;
References
public class primenumber1
{
    References
    public static void Main(string[] args)
    {
        int n, i, m = 0, flag = 0;
        Console.Write("Enter the Number to check Prime: ");
        n = int.Parse(Console.ReadLine());
        m = n / 2;
        for (i = 2; i <= m; i++)
        {
            if (n % i == 0)
            {
                Console.Write("Number is not Prime.");
                flag = 1;
                break;
            }
        }
        if (flag == 0)
            Console.Write("Number is Prime.");
    }
}
```

Output:

```
Enter the Number to check Prime: 4
Number is not Prime
```

Task-05: Write a program to print the Fibonacci series

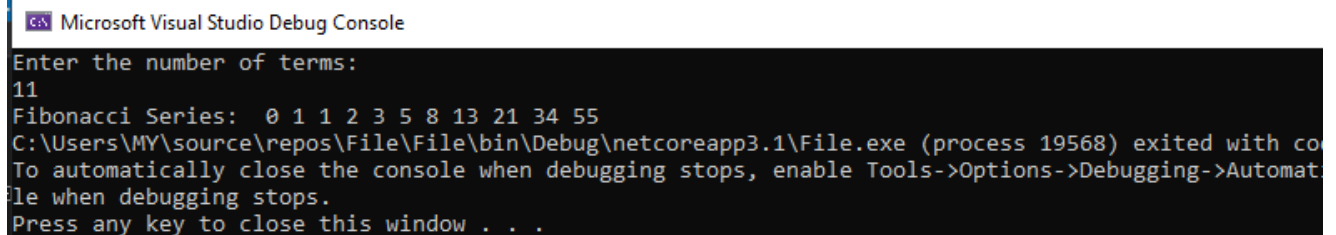
Program:

```

static void Main(string[] args)
{
    int i, First =0, Second=1, n ,next;
    Console.WriteLine("Enter the number of terms: ");
    n = int.Parse(Console.ReadLine());
    Console.Write("Fibonacci Series: ");
    for (i = 0; i < n; i++)
    {
        if (i <= 1)
        {
            next = i;
        }
        else
        {
            next = First + Second;
            First = Second;
            Second = next;
        }
        Console.Write(" " +next);
    }
}

```

Output:



Microsoft Visual Studio Debug Console

```

Enter the number of terms:
11
Fibonacci Series:  0 1 1 2 3 5 8 13 21 34 55
C:\Users\MY\source\repos\File\File\bin\Debug\netcoreapp3.1\File.exe (process 19568) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Task-06: Write a program to print the half pyramid pattern

Program:


```

using System;

namespace Patterns
{
    References
    class Program
    {
        References
        static void Main(string[] args)
        {
            //Pattern No : 01
            Console.WriteLine("Enter the number of rows: ");
            int n = int.Parse(Console.ReadLine());
            for(int i= 1; i <= n; i++)
            {
                Console.WriteLine();
                for(int j = 1; j <= i; j++)
                {
                    Console.Write(" * ");
                }
            }
        }
    }
}

```

Output:

```

Enter the number of rows:
7

 *
 * *
 * * *
 * * * *
 * * * * *
 * * * * * *
 * * * * * * *

```

C:\Users\MY\source\repos\Patterns\Patterns\bin\Debug\net6.0\Patterns.exe

Task-07: Write a program to print the half pyramid pattern with numbers

Program:

```
static void Main(string[] args)
{
    //Pattern No : 07
    Console.WriteLine("Enter the number of rows: ");
    int n = int.Parse(Console.ReadLine());
    int d=0;
    for (int i = 1; i <= n; i++)
    {
        for (int space = n - i; space > 0; space--)
        {
            Console.Write(" ");
        }
        for (int j = i; j < 2 * i; j++)
        {
            Console.Write(" " + j);
            d = j;
        }
        for (int b = d-1; b >= i; b--)
        {
            Console.Write(" " + b);
        }

        Console.WriteLine();
    }
}
```

Output:

Enter the number of rows:

5

```
      1
    2 3 2
  3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

C:\Users\MY\source\repos\Patterns\Patterns\Program.cs
To automatically close the console when debugging is complete, add the following line to the end of the file:

Task-08: Write a program to print the half pyramid inverse pattern

Program:

```
namespace Patterns
{
    References
    class Program
    {
        References
        static void Main(string[] args)
        {
            //Pattern No : 05
            Console.WriteLine("Enter the number of rows: ");
            int n = int.Parse(Console.ReadLine());
            for(int i= n; i >0; i--)
            {
                for (int space = 1; space<=n-i; space++)
                {
                    Console.Write(" ");
                }

                for (int j = i; j >0; j--)
                {
                    Console.Write("*");
                }
                Console.WriteLine();
            }
        }
    }
}
```

✓ No issues found

Output:

```
Enter the number of rows:  
6  
*****  
 *****  
  *****  
   *****  
    *****  
     *****  
      *****
```

Task-09: Write a program to print the pyramid pattern

Program:

```

References
static void Main(string[] args)
{
    //Pattern No : 06
    Console.WriteLine("Enter the number of rows: ");
    int n = int.Parse(Console.ReadLine());
    for (int i = 1; i <= n; i++)
    {
        for (int space = n - i; space > 0; space--)
        {
            Console.Write(" ");
        }
        for (int j = 1; j <= i; j++)
        {
            Console.Write("*");
        }
        for (int b = 1; b < i; b++)
        {
            Console.Write("*");
        }
        Console.WriteLine();
    }
}

```

Output:

```
Enter the number of rows:
6
    *
   ***
  *****
 *****
*****
*****
```

Task-10: Write a program to print the inverse pyramid pattern

Program:

```

class Program
{
    References
    static void Main(string[] args)
    {
        Console.WriteLine("Enter number of rows: ");
        int n = int.Parse(Console.ReadLine());
        for(int i = n; i > 0; i--)
        {
            for(int space = 1; space <= n - i; space++)
            {
                Console.Write(" ");
            }
            for(int a = 1; a <= i; a++)
            {
                Console.Write("* ");
            }
            for(int j = 0; j < i - 1; j++)
            {
                Console.Write("* ");
            }
            Console.WriteLine();
        }
    }
}

```

Output:

```

Enter number of rows:
5
* * * * *
 * * * *
  * * *
   * *
    *

```

Task-11: Write a program to print the diamond pattern

Program:

```

Console.WriteLine("Enter even number of rows: ");
int r = int.Parse(Console.ReadLine());
int n = (r / 2) + (r % 2);
for(int i = 1; i <=n; i++)
{
    for(int s = n-i; s>0; s--)
    {
        Console.Write(" ");
    }
    for(int j = 1; j <= i; j++)
    {
        Console.Write("*");
    }
    for(int b = 1; b < i; b++)
    {
        Console.Write("*");
    }Console.WriteLine();
}
for (int i = n-1; i > 0; i--)
{
    for(int s = 2; s <= n-i+1; s++)
    {
        Console.Write(" ");
    }
    for(int a = 1; a <= i; a++)
    {
        Console.Write("*");
    }
    for (int c = 0; c < i - 1; c++)
    {
        Console.Write("*");
    } Console.WriteLine();
}

```

Output:

```
Enter even number of rows:
7
  *
 ***
*****
*****
  *****
   ***
    *
```

C:\Users\MY\source\repos\cls\cls\bin\Debug\ne

Task-12: Write a program to print pascal's triangle

Program:


```

class Program
{
    //References
    static void Main(string[] args)
    {
        //pascal's triangle
        Console.WriteLine("Enter number of rows: ");
        int n = int.Parse(Console.ReadLine());
        for(int i = 0; i < n; i++)
        {
            for(int s = 1; s <= n - i; s++)
            {
                Console.Write(" ");
            }
            int k = 1;
            for(int j = 0; j <= i; j++)
            {
                Console.Write(" "+k);
                k = k * (i - j) / (j + 1);
            }

            Console.WriteLine();
        }
    }
}

```

Output:

```

Enter number of rows:
5

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

C:\Users\MY\source\repos\cls\cls\bin\Debug\netcor

```

Task-13: Write a program to compare two string without using string library functions

Program:

```
// compare two String
bool state = false;
String str1, str2;
int l1=0, l2=0;
Console.WriteLine("Enter First String: ");
str1 = Console.ReadLine();
Console.WriteLine("Enter Second String: ");
str2 = Console.ReadLine();
foreach(char i in str1)
{
    l1++;
}
foreach(char j in str2)
{
    l2++;
}
if (l1 == l2)
{
    for(int i = 0; i < l1; i++)
    {
        if (str1[i] == str2[i])
        {
            state = true;
        }
        else
        {
            state = false;
            break;
        }
    }
}
if (state == true)
{
    Console.WriteLine("Matched");
}
else
{
    Console.WriteLine("Not Matched");
}
```

Output:

```
Enter First String:
Sara # Thakur $ 27
Enter Second String:
Sara # Thakur $ 27
Matched

C:\Users\MY\source\repos\abc\abc\b
```

```
Enter First String:
abc
Enter Second String:
bca
Not Matched

C:\Users\MY\source\repos\abc\abc\b
```

Task-14: Write a program to count a total number of alphabets, digits and special characters in a string

Program:

```

using System;
References
public class Exercise7
{
    References
    public static void Main()
    {
        string str;
        int alp, digit, splch, i, l;
        alp = digit = splch = i = 0;
        Console.Write("Input the string : ");
        str = Console.ReadLine();
        l = str.Length;

        while (i < l)
        {
            if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))
            {
                alp++;
            }
            else if (str[i] >= '0' && str[i] <= '9')
            {
                digit++;
            }
            else
            {
                splch++;
            }

            i++;
        }

        Console.Write("Number of Alphabets in the string is : {0}\n", alp);
        Console.Write("Number of Digits in the string is : {0}\n", digit);
        Console.Write("Number of Special characters in the string is : {0}\n\n", splch);
    }
}

```

Output:

```

Input the string : rupali%2709
Number of Alphabets in the string is : 6
Number of Digits in the string is : 4
Number of Special characters in the string is : 1

```

Task-15: Write a program to copy one string into another string

Program:

```

//copy one string to another string
int l=0;
String str;
Console.WriteLine("Enter the String: ");
str = Console.ReadLine();
foreach(char i in str)
{
    l++;
}
String[] str1 = new string[l];
for (int i = 0; i < l; i++)
{
    String temp= str[i].ToString();
    str1[i] = temp;
}
Console.WriteLine("Here it is.....");
for(int i=0;i<l;i++)
{
    Console.Write(str1[i]);
}

```

Output:

```

Enter the String:
SaraThakur###....27....
Here it is.....
SaraThakur###....27....
C:\Users\MY\source\repos\abc\abc\bin\Debug\netco
To automatically close the console when debuggin
le when debugging stops.

```

Task-16: Write a program to find maximum occurring character in a string

Program:

```
// maximum occurring character in string
String str, a = null;
int l = 0, count = 0, count1 = 0;
Console.WriteLine("Enter String: ");
str = Console.ReadLine();
foreach(char i in str)
{
    l++;
}
for(int i = 0; i < l; i++)
{
    count = 0;
    for( int j = i; j < l; j++)
    {
        if (str[i] == str[j])
        {
            count++;
        }
    }
    if (count > count1)
    {
        count1 = count;
        a = str[i].ToString();
    }
}
Console.WriteLine("Max: " + a + " is occurring : " + count1+" times.");
```

Output:

```
Enter String:
sara was searching for something....
Max: s is occurring : 4 times.

C:\Users\MY\source\repos\abc\abc\bin\Debug\netcoreapp3.1\abc
```

Task-17: Write a program to check whether a given substring is present in the given string

Program:

```

using System;
public class substr
{
    public static void Main()
    {
        string str1,str2;
        bool m;
        Console.Write("Input the string : ");
        str1 = Console.ReadLine();

        Console.Write("Input the substring to search : ");
        str2 = Console.ReadLine();
        m=str1.Contains(str2);

        if (m)
            Console.Write("The substring exists in the string.\n\n");
        else
            Console.Write("The substring is not exists in the string. \n\n");
    }
}

```

Output:

```

Input the string : Devdutt Paddikal become the emerging player of ipl 2020
Input the substring to search : Paddikal
The substring exists in the string.

```

```

Input the string : Rupali Singh
Input the substring to search : SaraThakur
The substring is not exists in the string.

```

Task-18: Write a program for Abstraction

Program:

```
raction  Circle  draw()
1  using System;
   3 references
2  public abstract class Shape
3  {
   4 references
4  public abstract void draw();
5  }
   1 reference
6  public class Rectangle : Shape
7  {
   4 references
8  public override void draw()
9  {
10 Console.WriteLine("drawing rectangle...");
11 }
12 }
   1 reference
13 public class Circle : Shape
14 {
   4 references
15 public override void draw()
16 {
17 Console.WriteLine("drawing circle...");
18 }
19 }
   0 references
20 public class TestAbstract
21 {
   0 references
22 public static void Main()
23 {
24 Shape s;
25 s = new Rectangle();
26 s.draw();
27 s = new Circle();
28 s.draw();
29 }
30 }
```

Output:

```
Microsoft Visual Studio Debug Console
drawing rectangle...
drawing circle...
C:\Users\apn\AY\abs\HA\source\repos\abstraction\abstraction\bin
```

Task-19: Write a program for single inheritance

Program:


```
heritance TestInheritance2 Main(string[] args)
using System;
1 reference
public class Animal
{
    1 reference
    public void eat() { Console.WriteLine("Eating..."); }
}
2 references
public class Dog : Animal
{
    1 reference
    public void bark() { Console.WriteLine("Barking..."); }
}
0 references
class TestInheritance2
{
    0 references
    public static void Main(string[] args)
    {
        Dog d1 = new Dog();
        d1.eat();
        d1.bark();
    }
}
```

Output:

```
Eating...
Barking...
```

Task-20: Write a program for multilevel inheritance

Program:

```
multilevelinheritance.Son
using System;

namespace multilevelinheritance
{
    2 references
    class Son : Father
    {
        1 reference
        public void DisplayTwo()
        {
            Console.WriteLine("Son.. ");
        }
        0 references
        static void Main(string[] args)
        {
            Son s = new Son();
            s.Display();
            s.DisplayOne();
            s.DisplayTwo();
            Console.Read();
        }
    }
    1 reference
    class Grandfather
    {
        1 reference
        public void Display()
        {
            Console.WriteLine("Grandfather...");
        }
    }
    1 reference
    class Father : Grandfather
    {
        1 reference
        public void DisplayOne()
        {
            Console.WriteLine("Father...");
        }
    }
}
```

```
multilevelinheritance.Son
    s.DisplayTwo();
    Console.Read();
}
1 reference
class Grandfather
{
    1 reference
    public void Display()
    {
        Console.WriteLine("Grandfather...");
    }
}
1 reference
class Father : Grandfather
{
    1 reference
    public void DisplayOne()
    {
        Console.WriteLine("Father...");
    }
}
```

Output:

```
Grandfather...  
Father...  
Son..
```

Task-21: Write a program for multiple inheritance

Program:

```
MultipleInheritApplication.Calculation.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MultipleInheritApplication
{
    1 reference
    interface calc1
    {
        2 references
        int add(int a, int b);
    }
    1 reference
    interface calc2
    {
        2 references
        int sub(int x, int y);
    }
    1 reference
    interface calc3
    {
        2 references
        int mul(int r, int s);
    }
    1 reference
    interface calc4
    {
        2 references
        int div(int c, int d);
    }
    2 references
    class Calculation : calc1, calc2, calc3, calc4
    {
        public int result1;
        2 references
        public int add(int a, int b)
        {
            return result1 = a + b;
        }
    }
}
```

No issues found

```
eritance
MultipleInheritApplication.Calculation.Prograr
Main(string[] args)

2 references
public int sub(int x, int y)
{
    return result2 = x - y;
}
public int result3;
2 references
public int mul(int r, int s)
{
    return result3 = r * s;
}
public int result4;
2 references
public int div(int c, int d)
{
    return result4 = c / d;
}

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        Calculation c = new Calculation();
        c.add(8, 2);
        c.sub(20, 10);
        c.mul(5, 2);
        c.div(20, 10);
        Console.WriteLine("Multiple Inheritance concept Using Interfaces :\n ");
        Console.WriteLine("Addition: " + c.result1);
        Console.WriteLine("Substraction: " + c.result2);
        Console.WriteLine("Multiplication : " + c.result3);
        Console.WriteLine("Division: " + c.result4);
        Console.ReadKey();
    }
}
```

Output:

Microsoft Visual Studio Debug Console

```
Multiple Inheritance concept Using Interfaces :

Addition: 10
Substraction: 10
Multiplication :10
Division: 2
```

Task-22: Write a program for method overloading

Program:

```
using System;

namespace MyApplication
{
    0 references
    class Program
    {
        1 reference
        static int PlusMethod(int x, int y)
        {
            return x + y;
        }

        1 reference
        static double PlusMethod(double x, double y)
        {
            return x + y;
        }

        0 references
        static void Main(string[] args)
        {
            int myNum1 = PlusMethod(8, 5);
            double myNum2 = PlusMethod(4.3, 6.26);
            Console.WriteLine("Int: " + myNum1);
            Console.WriteLine("Double: " + myNum2);
        }
    }
}
```

Output:

```
Int: 13
Double: 10.559999999999999
```

Task-23: Write a program for method overriding

Program:

```
using System;
1 reference
public class Animal
{
    2 references
    public virtual void eat()
    {
        Console.WriteLine("Eating...");
    }
}
2 references
public class Dog : Animal
{
    2 references
    public override void eat()
    {
        Console.WriteLine("Eating bread...");
    }
}
0 references
public class TestOverriding
{
    0 references
    public static void Main()
    {
        Dog d = new Dog();
        d.eat();
    }
}
```

Output:

```
Eating bread...
```

Task-24: Write a program for interface

Program:

```
1 using System;
2
3 namespace MyApplication
4 {
5     1 reference
6     interface IFirstInterface
7     {
8         2 references
9         void myMethod(); // interface method
10    }
11    1 reference
12    interface ISecondInterface
13    {
14        2 references
15        void myOtherMethod(); // interface method
16    }
17    2 references
18    class DemoClass : IFirstInterface, ISecondInterface
19    {
20        2 references
21        public void myMethod()
22        {
23            Console.WriteLine("Some text..");
24        }
25        2 references
26        public void myOtherMethod()
27        {
28            Console.WriteLine("Some other text...");
29        }
30    }
31    0 references
32    class Program
33    {
34        0 references
35        static void Main(string[] args)
36        {
37            DemoClass myObj = new DemoClass();
38            myObj.myMethod();
39            myObj.myOtherMethod();
40        }
41    }
42 }
```

No issues found

Output:

```
Some text..
Some other text...
```

Task-25: Write a program for exception handling through try and catch

Program:


```
catch | tryandcatch.Program
using System;

namespace tryandcatch
{
    References
    class Program
    {
        References
        static void Main(string[] args)
        {
            try
            {
                int[] myNumbers = { 1, 2, 3 };
                Console.WriteLine(myNumbers[10]);
            }
            catch (Exception e)
            {
                Console.WriteLine("Something went wrong.");
            }
        }
    }
}
```

Output:

```
Microsoft Visual Studio Debug Console
Something went wrong.
```

Task-26: Write a program for properties

Program:

```

using System;
2 references
public class Employee
{
    private string name;

    2 references
    public string Name
    {
        get
        {
            return name;
        }
        set
        {
            name = value;
        }
    }
}
0 references
class TestEmployee{
    0 references
    public static void Main(string[] args)
    {
        Employee e1 = new Employee();
        e1.Name = "sajal laddha";
        Console.WriteLine("Employee Name: " + e1.Name);
    }
}

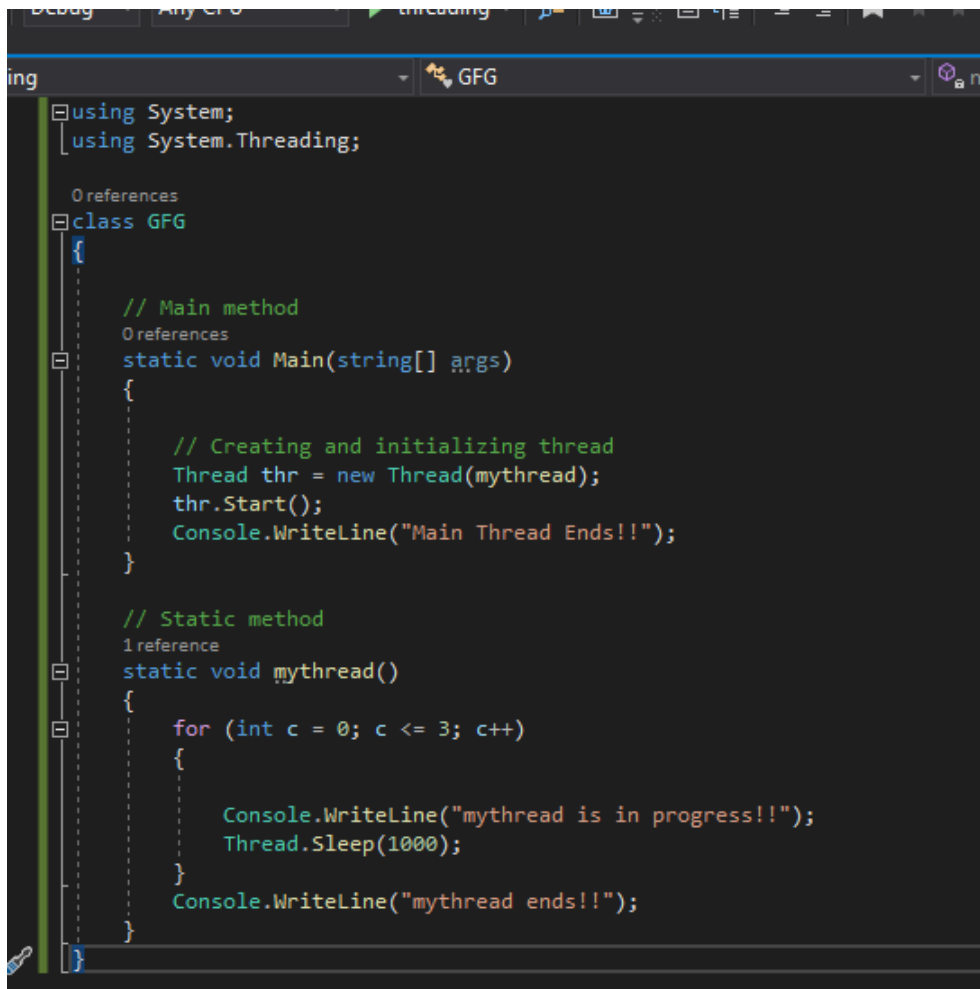
```

Output:

Employee Name: Rupali Singh

Task-27: Write a program for threading

Program:

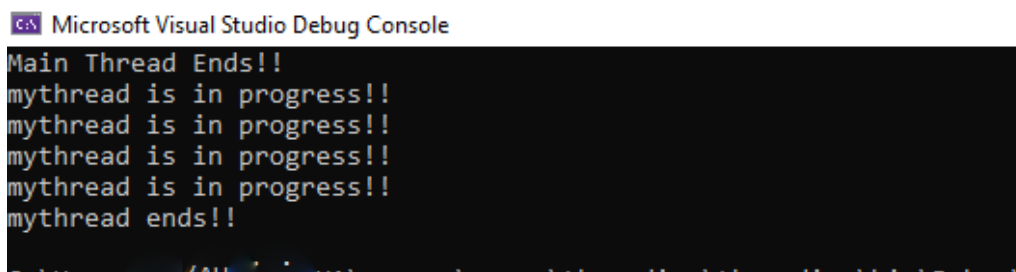


```
using System;
using System.Threading;

class GFG
{
    // Main method
    static void Main(string[] args)
    {
        // Creating and initializing thread
        Thread thr = new Thread(mythread);
        thr.Start();
        Console.WriteLine("Main Thread Ends!!");
    }

    // Static method
    static void mythread()
    {
        for (int c = 0; c <= 3; c++)
        {
            Console.WriteLine("mythread is in progress!!");
            Thread.Sleep(1000);
        }
        Console.WriteLine("mythread ends!!");
    }
}
```

Output:



```
Microsoft Visual Studio Debug Console
Main Thread Ends!!
mythread is in progress!!
mythread is in progress!!
mythread is in progress!!
mythread is in progress!!
mythread ends!!
```

Task-28: Write a program to access data from database using ADO.NET

Program:

```

using System;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace WindowsApplication1
{
    1 reference
    public partial class Form1 : Form
    {
        0 references
        public Form1()
        {
            InitializeComponent();
        }
        0 references
        private void button1_Click(object sender, EventArgs e)
        {
            string connetionString = null;
            SqlConnection connection;
            SqlCommand command;
            string sql = null;
            SqlDataReader dataReader;
            connetionString = "Data Source=ServerName;Initial Catalog=DatabaseName;User ID=UserName;Password=Password";
            sql = "Your SQL Statement Here , like Select * from product";
            connection = new SqlConnection(connetionString);
            try
            {
                connection.Open();
                command = new SqlCommand(sql, connection);
                dataReader = command.ExecuteReader();
                while (dataReader.Read())
                {
                    MessageBox.Show(dataReader.GetValue(0) + " - " + dataReader.GetValue(1) + " - " + dataReader.GetValue(2));
                }
                dataReader.Close();
                command.Dispose();
                connection.Close();
            }
        }
    }
}

```

```

SqlCommand command;
string sql = null;
SqlDataReader dataReader;
connetionString = "Data Source=ServerName;Initial Catalog=DatabaseName;User ID=UserName;Password=Password";
sql = "Your SQL Statement Here , like Select * from product";
connection = new SqlConnection(connetionString);
try
{
    connection.Open();
    command = new SqlCommand(sql, connection);
    dataReader = command.ExecuteReader();
    while (dataReader.Read())
    {
        MessageBox.Show(dataReader.GetValue(0) + " - " + dataReader.GetValue(1) + " - " + dataReader.GetValue(2));
    }
    dataReader.Close();
    command.Dispose();
    connection.Close();
}
catch (Exception ex)
{
    MessageBox.Show("Can not open connection ! ");
}
}
}

```

Task-29: Write a program for namespace

Program:

```
pace | first_space.namespace_cl | func()
using System;

namespace first_space
{
    2 references
    class namespace_cl
    {
        1 reference
        public void func()
        {
            Console.WriteLine("Inside first_space");
        }
    }
}

namespace second_space
{
    2 references
    class namespace_cl
    {
        1 reference
        public void func()
        {
            Console.WriteLine("Inside second_space");
        }
    }
}

0 references
class TestClass
{
    0 references
    static void Main(string[] args)
    {
        first_space.namespace_cl fc = new first_space.namespace_cl();
        second_space.namespace_cl sc = new second_space.namespace_cl();
        fc.func();
        sc.func();
        Console.ReadKey();
    }
}
```

```
pace | first_space.namespace_cl | func()
namespace second_space
{
    2 references
    class namespace_cl
    {
        1 reference
        public void func()
        {
            Console.WriteLine("Inside second_space");
        }
    }
}

0 references
class TestClass
{
    0 references
    static void Main(string[] args)
    {
        first_space.namespace_cl fc = new first_space.namespace_cl();
        second_space.namespace_cl sc = new second_space.namespace_cl();
        fc.func();
        sc.func();
        Console.ReadKey();
    }
}
```

Task-30: Write a program to print square pattern

Program:

```
int n = int.Parse(Console.ReadLine());
for(int r = n; r >= 1; r--)
{
    for(int i = n; i > r; i--)
    {
        Console.Write(" "+i);
    }
    for (int j = 1; j <= 2*r-1 ; j++)
    {
        Console.Write(" "+r);
    }
    for(int k = r + 1; k <= n; k++)
    {
        Console.Write(" "+k);
    }
    Console.WriteLine();
}
for(int r = 2; r <=n; r++)
{
    for (int i = n; i > r; i--)
    {
        Console.Write(" "+i);
    }
    for (int j = 1; j <= 2 * r - 1; j++)
    {
        Console.Write(" "+r);
    }
    for (int k = r + 1; k <= n; k++)
    {
        Console.Write(" "+k);
    }
    Console.WriteLine();
}
```

Output:

Enter the number:

6

```
6 6 6 6 6 6 6 6 6 6 6
6 5 5 5 5 5 5 5 5 5 6
6 5 4 4 4 4 4 4 4 5 6
6 5 4 3 3 3 3 3 4 5 6
6 5 4 3 2 2 2 3 4 5 6
6 5 4 3 2 1 2 3 4 5 6
6 5 4 3 2 2 2 3 4 5 6
6 5 4 3 3 3 3 3 4 5 6
6 5 4 4 4 4 4 4 4 5 6
6 5 5 5 5 5 5 5 5 5 6
6 6 6 6 6 6 6 6 6 6 6
```