

Dev Sansakriti Vishwavidyalaya Haridwar



Assignment

On

C#.NET

Session 2020-2021

Submitted to:

Mr. Chandrashekhar Patel

Dept. of computer science

Submitted by:

Shail Kumari Sahu

BCA 5th semester

Department of computer science DSVV

Gayatrikunj Shantikunj Haridwar

s.no.	Task	Page no.	Remark
01	write a program to print Armstrong numbers	01	
02	write a program to print factorial of a number	01	
03	Write a program to find GCD of two number	02	
04	Write a program to check if a number is prime number	02	
05	Write a program to print the Fibonacci series	03	
06	Write a program to print half pyramid pattern	03	
07	Write a program to print half pyramid pattern with numbers	04	
08	Write a program to print half pyramid inverse pattern	05	
09	Write a program to print pyramid pattern	06	
10	Write a program to print inverse pyramid pattern	07	
11	Write a program to print diamond pattern	08	
12	Write a program to print Pascal's triangle	09	
13	Write a program to compare two string without using string library functions	09-10	
14	Write a program to count a total number of alphabets, digits and special characters in a string	11	
15	Write a program to copy one string another string	12	
16	Write a program to find maximum occurring character in a string	12	
17	Write a program to check whether a given substring is present in the given string	13	
18	Write a program for Abstraction	14	
19	Write a program for single inheritance	15	
20	Write a program to multilevel inheritance	16	
21	Write a program to multiple inheritance	17	
22	Write a program for method overloading	18	
23	Write a program for method overriding	19	
24	Write a program for interface	20	
25	Write a program to for exception handling through try and catch	21	
26	Write a program for properties	21	
27	Write a program for Threading	22	
28	Write a program to access data from database using ADO.NET	23	
29	Write a program for namespace	24	

Signature.....

Task 1 Write a program to check Armstrong Number

The screenshot shows a Visual Studio window with a C# file named `Program.cs` in the `Armstrong` namespace. The code defines a `Program` class with a `Main` method that takes an array of strings as input. It prompts the user to enter a number, parses it, and then checks if it is an Armstrong number by calculating the sum of the cubes of its digits. The debug console shows the output for the input 123, which is not an Armstrong number.

```

1 using System;
2
3 namespace Armstrong
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int n, r, sum = 0, temp;
10            Console.WriteLine("Enter the Number= ");
11            n = int.Parse(Console.ReadLine());
12            temp = n;
13            while (n > 0)
14            {
15                r = n % 10;
16                sum = sum + (r * r * r);
17                n = n / 10;
18            }
19            if (temp == sum)
20                Console.WriteLine("Armstrong Number.");
21            else
22                Console.WriteLine("Not Armstrong Number.");
23        }
24    }
25 }

```

Microsoft Visual Studio Debug Console

```

Enter the Number= 123
Not Armstrong Number.
C:\Users\HP\source\repos\Armstrong\Armstrong\bin\Debug\netcoreapp3.1\Armstrong.exe (process 4144) 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 2 Write a program to find Factorial of a number

The screenshot shows a Visual Studio window with a C# file named `Program.cs` in the `FactorialEx` namespace. The code defines a `Program` class with a `Main` method that takes an array of strings as input. It prompts the user to enter a number, parses it, and then calculates the factorial using a for loop. The debug console shows the output for the input 5, which is 120.

```

1 using System;
2
3 namespace FactorialEx
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int i, fact = 1, number;
10            Console.WriteLine("Enter any Number: ");
11            number = int.Parse(Console.ReadLine());
12            for (i = 1; i <= number; i++)
13            {
14                fact = fact * i;
15            }
16            Console.WriteLine("Factorial of " + number + " is: " + fact);
17        }
18    }
19 }
20

```

Microsoft Visual Studio Debug Console

```

Enter any Number: 5
Factorial of 5 is: 120
C:\Users\HP\source\repos\FactorialEx\FactorialEx\bin\Debug\netcoreapp3.1\FactorialEx.exe (process 15752) 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 3 Write a program to find GCD of two number

The screenshot shows a C# program in Visual Studio. The code is as follows:

```

7  static void Main(string[] args)
8  {
9      Console.WriteLine("enter two number");
10     int a = int.Parse(Console.ReadLine());
11     int b = int.Parse(Console.ReadLine());
12     int temp, remainder;
13     if(b>a)
14     {
15         temp = b;
16         b = a;
17         a = temp;
18     }
19     while(a!=0&&b!=0)
20     {
21         remainder = a % b;
22         a = b;
23         b = remainder;
24     }
25     if(a==0&&b!=0)
26     {
27         Console.WriteLine("GCD:" + b);
28     }
29     else if(a!=0&&b==0)
30     {
31         Console.WriteLine("GCD:" + a);
32     }
33 }

```

The debug console shows the following output:

```

enter two number
192
270
GCD:6

C:\Users\HP\source\repos\GCD\
process 17092) exited with co
To automatically close the co

```

Task 4 Write a program to Prime number.

The screenshot shows a C# program in Visual Studio. The code is as follows:

```

2  namespace prime
3  {
4      0 references
5      class Program
6      {
7          0 references
8          static void Main(string[] args)
9          {
10             int n, i, m = 0, flag = 0;
11             Console.Write("Enter the Number to check Prime: ");
12             n = int.Parse(Console.ReadLine());
13             m = n / 2;
14             for (i = 2; i <= m; i++)
15             {
16                 if (n % i == 0)
17                 {
18                     Console.Write("Number is not Prime.");
19                     flag = 1;
20                     break;
21                 }
22             }
23             if (flag == 0)
24                 Console.Write("Number is Prime.");
25         }
26     }
27 }

```

The debug console shows the following output:

```

Enter the Number to check Prime: 453
Number is not Prime.
C:\Users\HP\source\repos\prime\prime\bin\Debug\netcor
eapp3.1\prime.exe (process 11872) exited with code 0.

To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically c

```

Task 5 Write a program to print Fibonacci series

The screenshot shows the Visual Studio IDE with a C# program named `FibonacciEx`. The code is as follows:

```

6  {
7      0 references
8      static void Main(string[] args)
9      {
10         int n1 = 0, n2 = 1, n3, i, number;
11         Console.WriteLine("Enter the number of elements: ");
12         number = int.Parse(Console.ReadLine());
13         Console.WriteLine(n1 + " " + n2 + " "); //printing 0 and 1
14         for (i = 2; i < number; ++i) //loop starts from 2 because 0 and 1 are already printed
15         {
16             n3 = n1 + n2;
17             Console.WriteLine(n3 + " ");
18             n1 = n2;
19             n2 = n3;
20         }
21     }
22 }
23

```

The output window shows the following text:

```

Enter the number of elements: 10
0 1 1 2 3 5 8 13 21 34
C:\Users\HP\source\repos\FibonacciEx\FibonacciE
x\bin\Debug\netcoreapp3.1\FibonacciEx.exe (proc
ess 19232) exited with code 0.
To automatically close the console when debuggi
ng stops, enable Tools->Options->Debugging->Aut
omatically close the console when debugging sto
ps.

```

Task 6 Write a program to print the half pyramid pattern.

The screenshot shows the Visual Studio IDE with a C# program named `pyramid`. The code is as follows:

```

7  static void Main(string[] args)
8  {
9      int space, rows;
10     Console.WriteLine("enter the number of rows:");
11     rows = int.Parse(Console.ReadLine());
12     for (int i = 0; i <= rows; i++)
13     {
14         for (int star = 0; star < i; star++)
15         {
16             Console.WriteLine("*");
17         }
18         for (space = i; space < rows; space++)
19         {
20             Console.Write(" ");
21         }
22         Console.WriteLine();
23     }
24 }

```

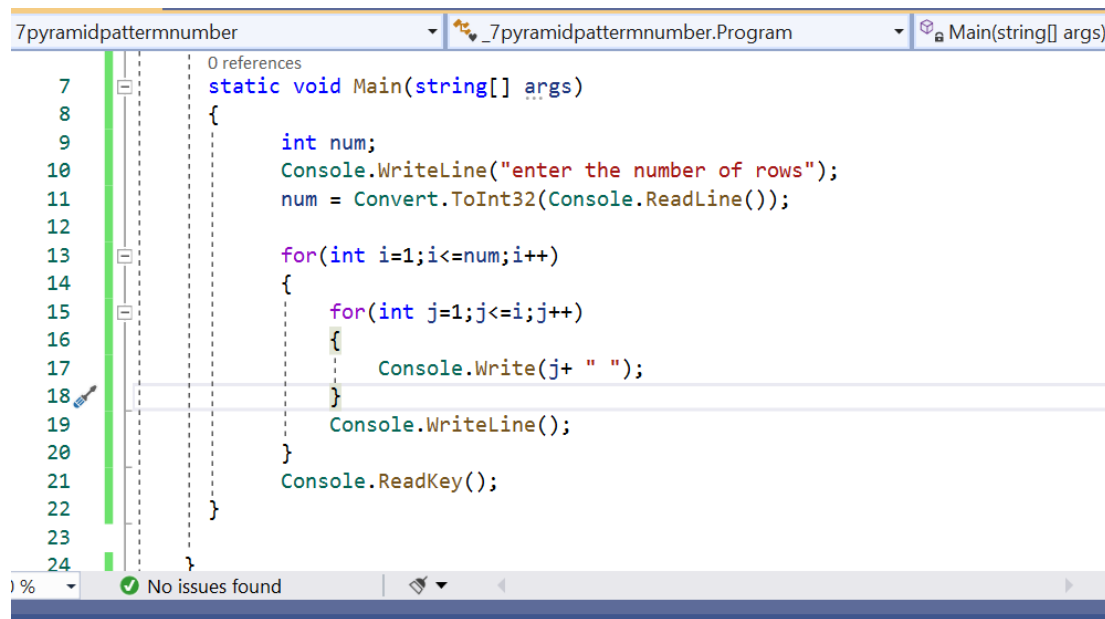
Output:-

```

Enter the number of rows:
6
*
**
***
****
*****
*****
Press any key to continue...

```

Task 7 Write a program to print the half pyramid pattern with numbers.

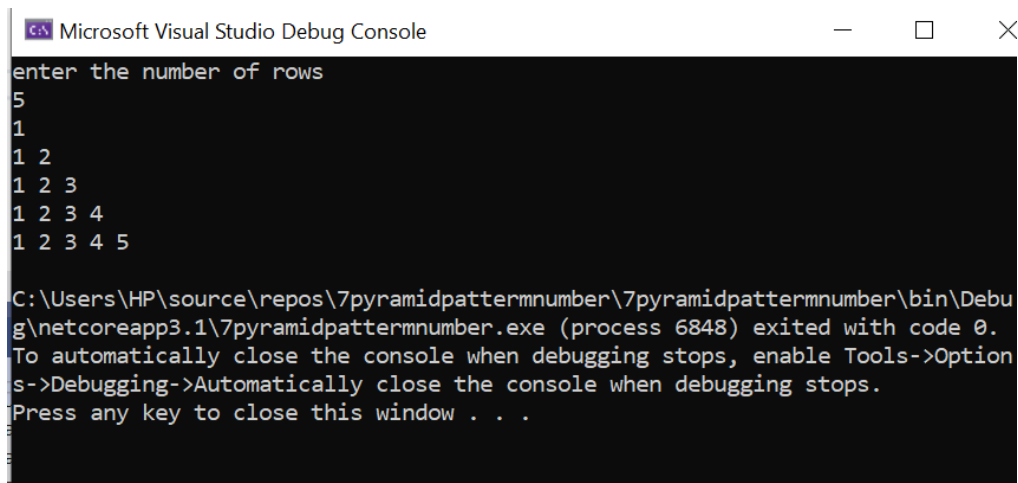


```
7 pyramidpatternnumber
8 static void Main(string[] args)
9 {
10     int num;
11     Console.WriteLine("enter the number of rows");
12     num = Convert.ToInt32(Console.ReadLine());
13     for(int i=1;i<=num;i++)
14     {
15         for(int j=1;j<=i;j++)
16         {
17             Console.Write(j+ " ");
18         }
19         Console.WriteLine();
20     }
21     Console.ReadKey();
22 }
23
24
```

0 references

1 % No issues found

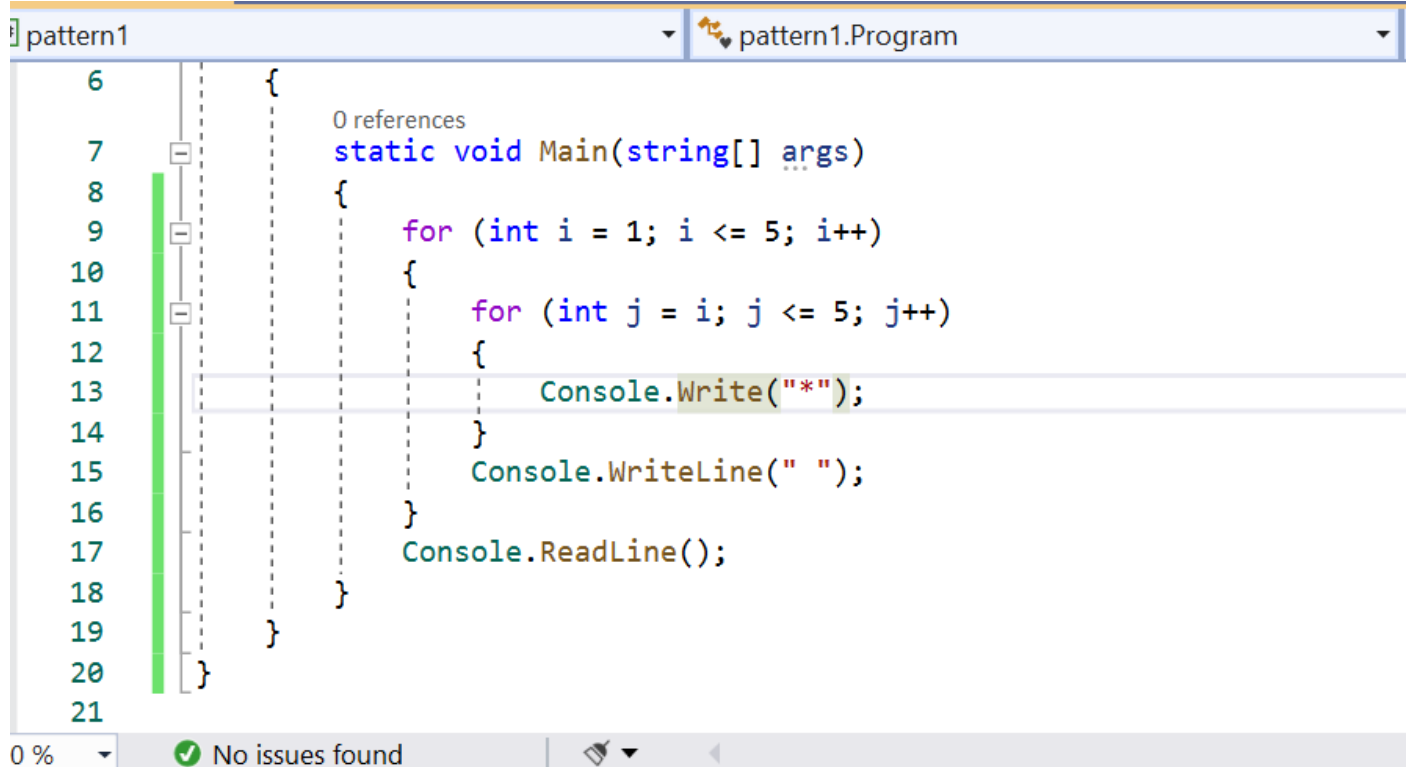
Output:-



```
C:\Users\HP\source\repos\7pyramidpatternnumber\7pyramidpatternnumber\bin\Debug\netcoreapp3.1\7pyramidpatternnumber.exe (process 6848) 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 . . .
```

enter the number of rows
5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

Task 8 Write a program to print the half pyramid inverse pattern

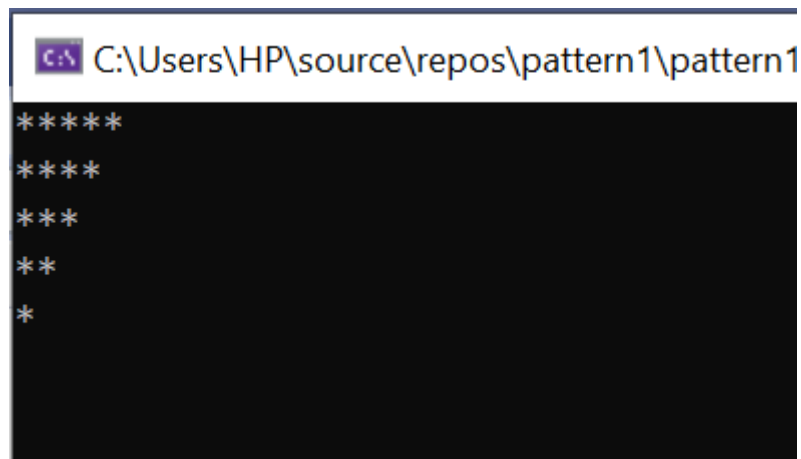


The screenshot shows a C# program in Visual Studio. The file explorer on the left shows a project named 'pattern1' with a file 'pattern1.Program'. The code editor displays the following C# code:

```
6 {  
7     0 references  
8     static void Main(string[] args)  
9     {  
10         for (int i = 1; i <= 5; i++)  
11         {  
12             for (int j = i; j <= 5; j++)  
13             {  
14                 Console.Write("*");  
15             }  
16             Console.WriteLine(" ");  
17         }  
18     }  
19 }  
20 }  
21 }
```

The status bar at the bottom indicates '0 %' and 'No issues found'.

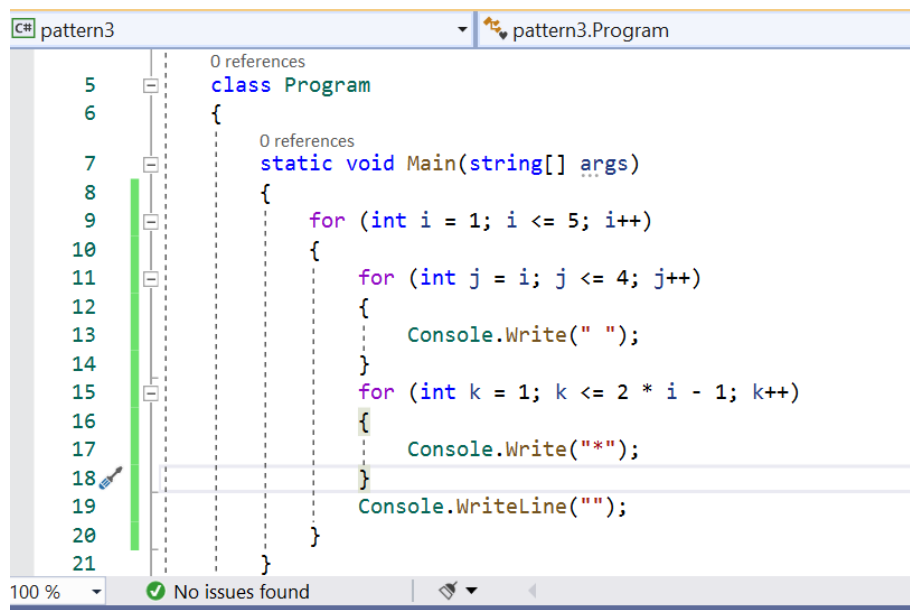
Output:-



The screenshot shows a console window with the following output:

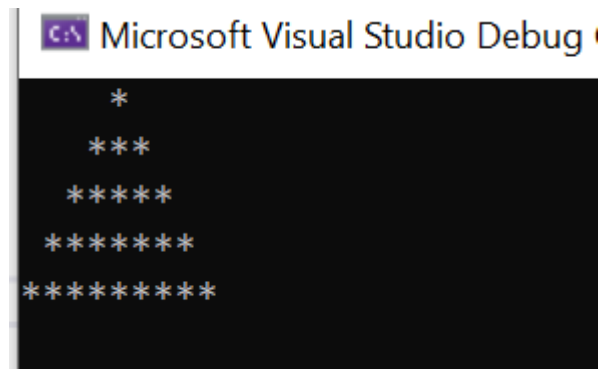
```
C:\Users\HP\source\repos\pattern1\pattern1  
*****  
****  
***  
**  
*
```

Task 9 Write a program to print pyramid pattern.



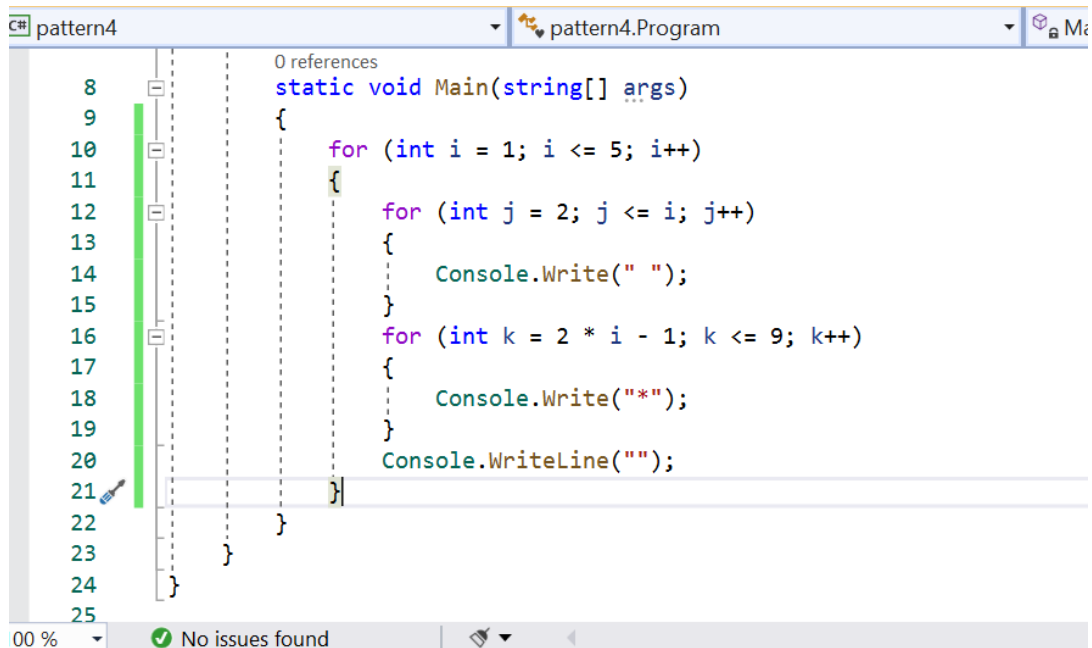
```
5  class Program
6  {
7      0 references
8      static void Main(string[] args)
9      {
10         for (int i = 1; i <= 5; i++)
11         {
12             for (int j = i; j <= 4; j++)
13             {
14                 Console.Write(" ");
15             }
16             for (int k = 1; k <= 2 * i - 1; k++)
17             {
18                 Console.Write("*");
19             }
20             Console.WriteLine("");
21         }
22     }
```

Output:-



```
C:\> Microsoft Visual Studio Debug
*
***
*****
*****
*****
```


Task 10 Write a program to print inverse pyramid pattern.



The screenshot shows a C# program in Visual Studio Code. The file is named 'pattern4' and the program is 'pattern4.Program'. The code is as follows:

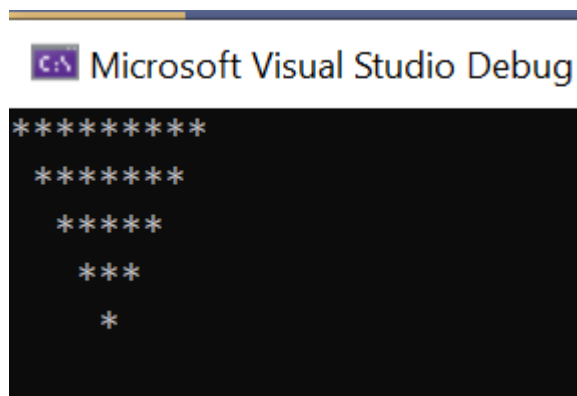
```

0 references
static void Main(string[] args)
{
    for (int i = 1; i <= 5; i++)
    {
        for (int j = 2; j <= i; j++)
        {
            Console.Write(" ");
        }
        for (int k = 2 * i - 1; k <= 9; k++)
        {
            Console.Write("*");
        }
        Console.WriteLine("");
    }
}

```

The code uses nested loops to print an inverse pyramid pattern. The outer loop iterates from i = 1 to i = 5. The inner loop for spaces iterates from j = 2 to j = i. The inner loop for asterisks iterates from k = 2 * i - 1 to k = 9. The pattern is printed line by line using Console.WriteLine("").

Output:-



The screenshot shows the Microsoft Visual Studio Debug console. The output of the program is an inverse pyramid pattern of asterisks:

```

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

```

Task 11 Write a program for print the diamond pattern.

```

9      {
10     for(int i=1;i<=5;i++)
11     {
12         for (int j = i; j <= 4; j++)
13         {
14             Console.Write(" ");
15         }
16         for (int k = 1; k <= 2 * i - 1; k++)
17         {
18             Console.Write("*");
19         }
20         Console.WriteLine();
21     }
22     for(int i=2;i<=5;i++)
23     {
24         for (int j = 2; j <= i; j++)
25         {
26             Console.Write(" ");
27         }
28         for(int k=2*i-1;k<=9;k++)
29         {
30             Console.Write("*");
31         }
32         Console.WriteLine();
33     }
34 }
  
```

Output:-

```

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

C:\Users\HP\source\repo
To automatically close
  
```

Task 12 Write the program for Pascal's triangle.

The screenshot shows a C# program in Visual Studio. The code is as follows:

```

8  {
9      Console.WriteLine("enter the number of rows");
10     int n = int.Parse(Console.ReadLine());
11     for(int i=0;i<n;i++)
12     {
13         for(int p=1;p<=n-i;p++)
14         {
15             Console.Write(" ");
16         }
17         int k = 1;
18         for(int j=0;j<=i;j++)
19         {
20             Console.Write(" " + k);
21             k = k * (i - j) / (j + 1);
22         }
23         Console.WriteLine();
24     }
25 }

```

The output window shows the result of running the program with 5 rows:

```

enter the number of rows
5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

The status bar indicates "No issues found".

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

The screenshot shows a C# program in Visual Studio. The code is as follows:


```

1  using System;
2  public class _13str
3  {
4      public static void Main()
5      {
6          string str1, str2;
7          int flg = 0;
8          int i = 0, l1, l2, yn = 0;
9          Console.Write("input the 1st string:");
10         str1 = Console.ReadLine();
11         Console.Write("input the 2nd string:");
12         str2 = Console.ReadLine();
13         l1 = str1.Length;
14         l2 = str2.Length;
15         if(l1==l2)
16         {
17             for(i=0;i<l1;i++)
18             {
19                 if(str1[i]!=str2[i])
20                 {
21                     yn = 1;
22                     i = 11;
23                 }
24             }
25         }
26         if(l1==l2)
27         flg=0;
28         else if(l1>l2)
29         flg=1;

```

The status bar indicates "No issues found".

Output:-

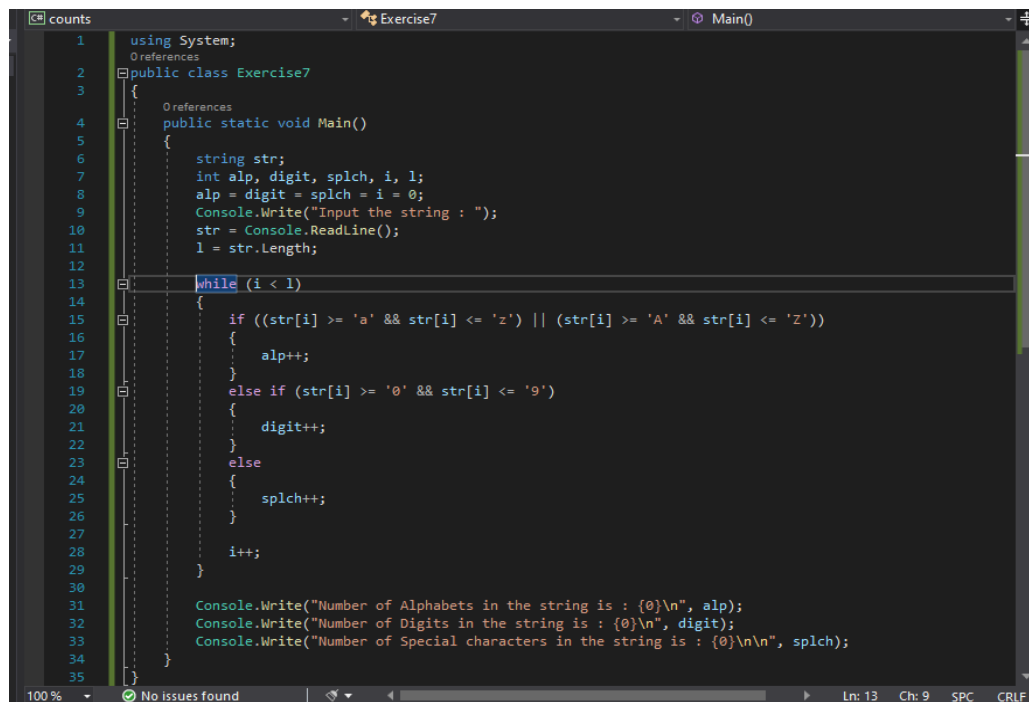
 Microsoft Visual Studio Debug Console

```
input the 1st string:shail
input the 2nd string:sonali

the length of the first string is smaller than second.

C:\Users\HP\source\repos\13str\13str\bin\Debug\netcoreapp
To automatically close the console when debugging stops,
le when debugging stops.
Press any key to close this window . . .
```

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



```

1  using System;
2  References
3  public class Exercise7
4  {
5      References
6      public static void Main()
7      {
8          string str;
9          int alp, digit, splch, i, l;
10         alp = digit = splch = i = 0;
11         Console.WriteLine("Input the string : ");
12         str = Console.ReadLine();
13         l = str.Length;
14         while (i < l)
15         {
16             if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))
17             {
18                 alp++;
19             }
20             else if (str[i] >= '0' && str[i] <= '9')
21             {
22                 digit++;
23             }
24             else
25             {
26                 splch++;
27             }
28             i++;
29         }
30
31         Console.WriteLine("Number of Alphabets in the string is : {0}\n", alp);
32         Console.WriteLine("Number of Digits in the string is : {0}\n", digit);
33         Console.WriteLine("Number of Special characters in the string is : {0}\n\n", splch);
34     }
35 }

```

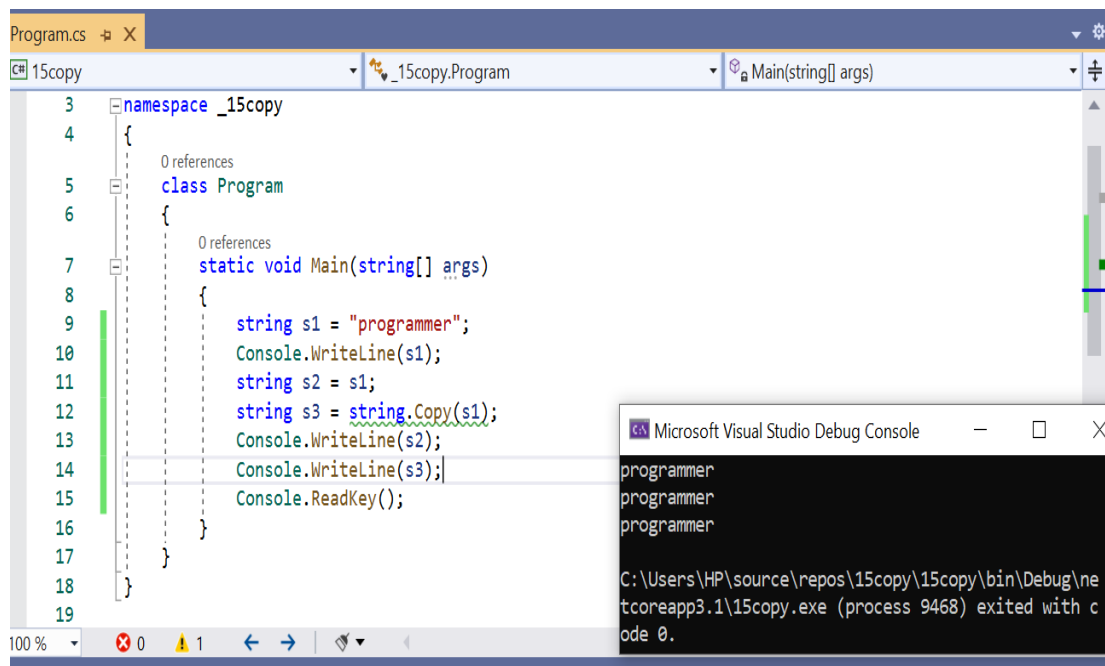
Microsoft Visual Studio Debug Console

```

Input the string : sajal@123
Number of Alphabets in the string is : 5
Number of Digits in the string is : 3
Number of Special characters in the string is : 1
C:\Users\SANJAY LADHA\source\repos\counts\counts\bin\Debug\netcon

```

Task 15 Write a program to copy one string to another string.



```

Program.cs
15copy
namespace _15copy
{
    class Program
    {
        static void Main(string[] args)
        {
            string s1 = "programmer";
            Console.WriteLine(s1);
            string s2 = s1;
            string s3 = string.Copy(s1);
            Console.WriteLine(s2);
            Console.WriteLine(s3);
            Console.ReadKey();
        }
    }
}

```

Microsoft Visual Studio Debug Console

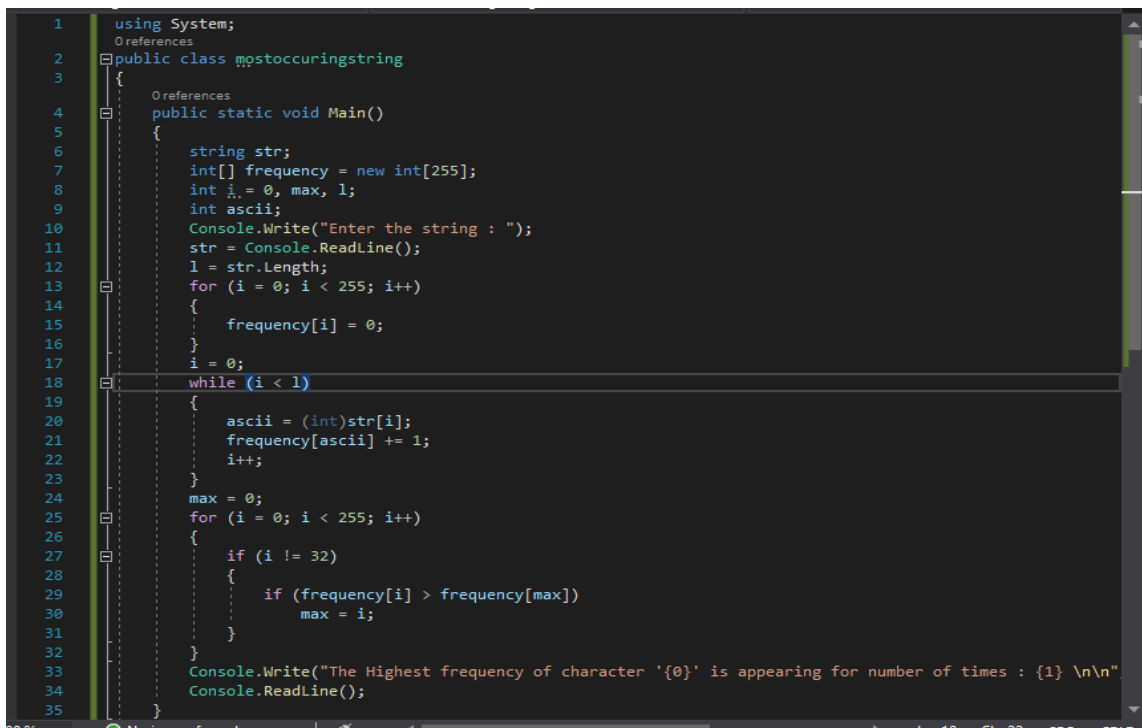
```

programmer
programmer
programmer

C:\Users\HP\source\repos\15copy\15copy\bin\Debug\netcoreapp3.1\15copy.exe (process 9468) exited with code 0.

```

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



```

1 using System;
2 public class mostoccurringstring
3 {
4     public static void Main()
5     {
6         string str;
7         int[] frequency = new int[255];
8         int i = 0, max = 1;
9         int ascii;
10        Console.Write("Enter the string : ");
11        str = Console.ReadLine();
12        l = str.Length;
13        for (i = 0; i < 255; i++)
14        {
15            frequency[i] = 0;
16        }
17        i = 0;
18        while (i < l)
19        {
20            ascii = (int)str[i];
21            frequency[ascii] += 1;
22            i++;
23        }
24        max = 0;
25        for (i = 0; i < 255; i++)
26        {
27            if (i != 32)
28            {
29                if (frequency[i] > frequency[max])
30                    max = i;
31            }
32        }
33        Console.Write("The Highest frequency of character '{0}' is appearing for number of times : {1} \n\n");
34        Console.ReadLine();
35    }
}

```

Output:-

```

C:\Users\SANJAY LADDHA\source\repos\mostoccurring\mostoccurring\bin\Debug\netcoreapp3.1\mostoccurring.exe
Enter the string : sajal laddha
The Highest frequency of character 'a' is appearing for number of times : 4

```

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

```

1  using System;
2  public class checksubstring
3  {
4      public static void Main()
5      {
6          string str1, str2;
7          bool m;
8
9
10         Console.Write("Input the string : ");
11         str1 = Console.ReadLine();
12
13         Console.Write("Input the substring to search : ");
14         str2 = Console.ReadLine();
15         m = str1.Contains(str2); // boolean value tapped here
16         if (m) // check boolean value is true or false.
17             Console.WriteLine("The substring exists in the string.\n\n");
18         else
19             Console.WriteLine("The substring is not exists in the string. \n\n");
20     }
21 }

```

Output:-

(substring exists)

```

Microsoft Visual Studio Debug Console
Input the string : sajal sanjay laddha
Input the substring to search : sanjay
The substring exists in the string.
C:\Users\SANJAY LADDHA\source\repos\checksubstring\checksubstring\bin\Debug\netcoreapp3.1\checksubstring.exe (process 13264) exited with code 0.

```

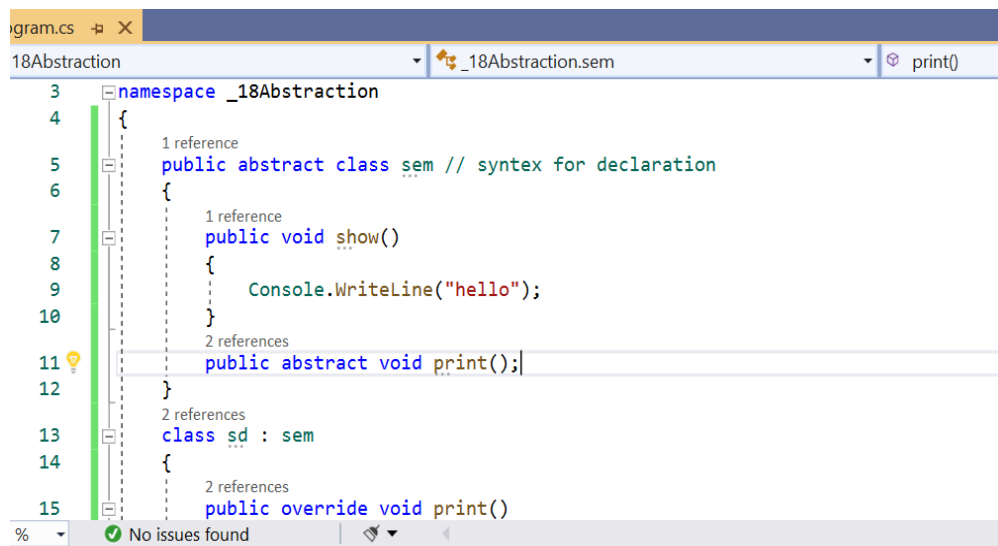
(substring doesn't exists)

```

Microsoft Visual Studio Debug Console
Input the string : sajal laddha
Input the substring to search : sanjaY
The substring is not exists in the string.
C:\Users\SANJAY LADDHA\source\repos\checksubstring\check

```

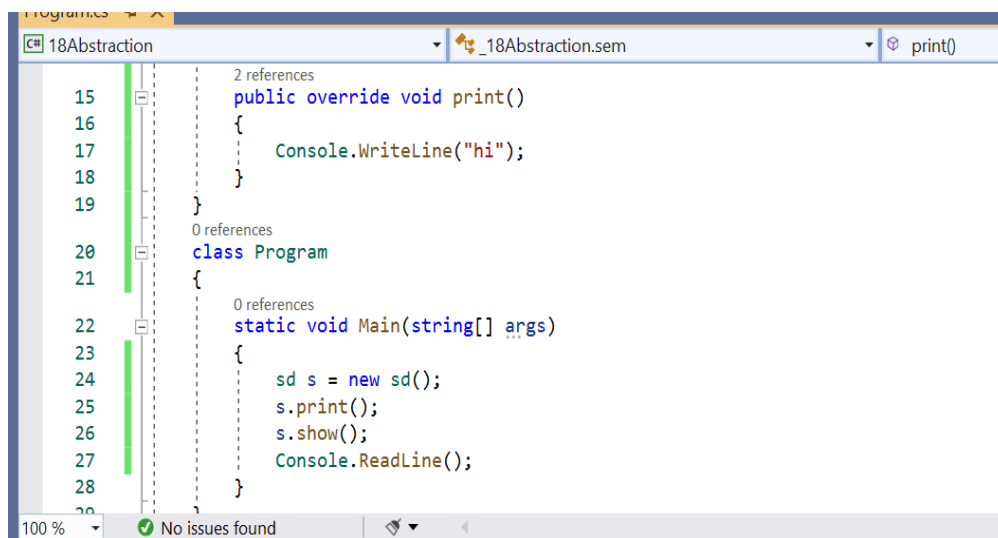
Task 18 Write a program for Abstraction.



```

3 namespace _18Abstraction
4 {
5     1 reference
6     public abstract class sem // syntax for declaration
7     {
8         1 reference
9         public void show()
10        {
11            Console.WriteLine("hello");
12        }
13        2 references
14        public abstract void print();
15    }
16    2 references
17    class sd : sem
18    {
19        2 references
20        public override void print()
21    }
22 }

```

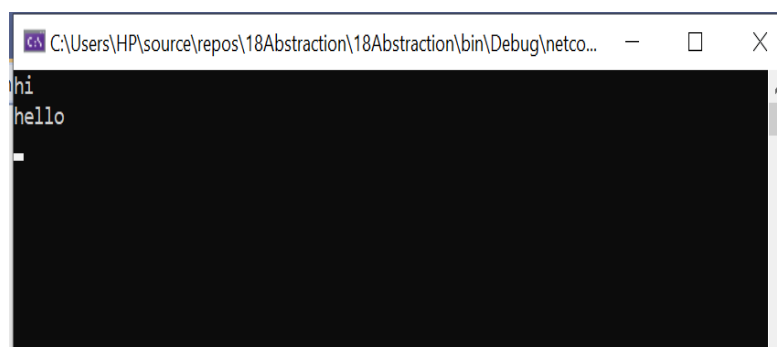


```

15 public override void print()
16 {
17     Console.WriteLine("hi");
18 }
19 }
20 0 references
21 class Program
22 {
23     0 references
24     static void Main(string[] args)
25     {
26         sd s = new sd();
27         s.print();
28         s.show();
29         Console.ReadLine();
30     }
31 }

```

Output:-

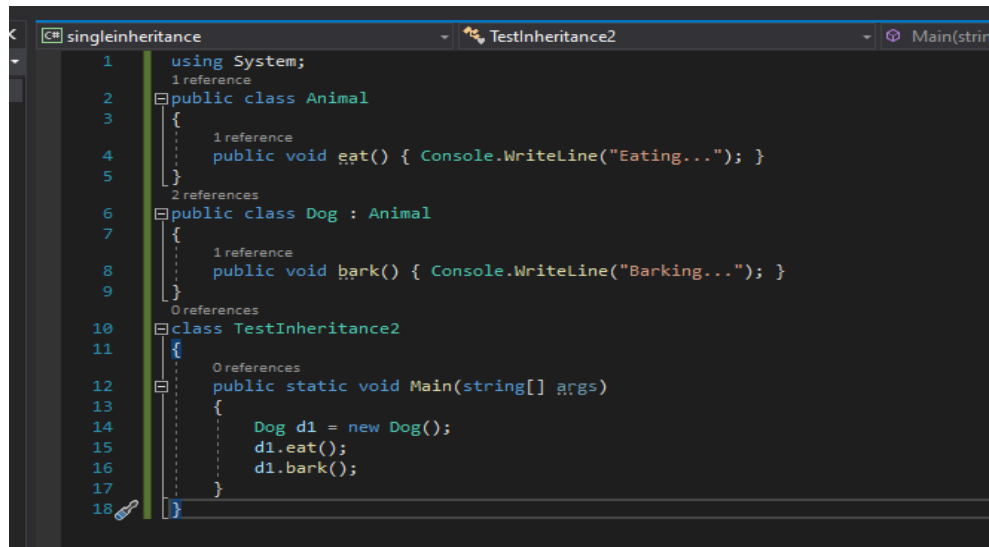


```

C:\Users\HP\source\repos\18Abstraction\18Abstraction\bin\Debug\netco...
hi
hello

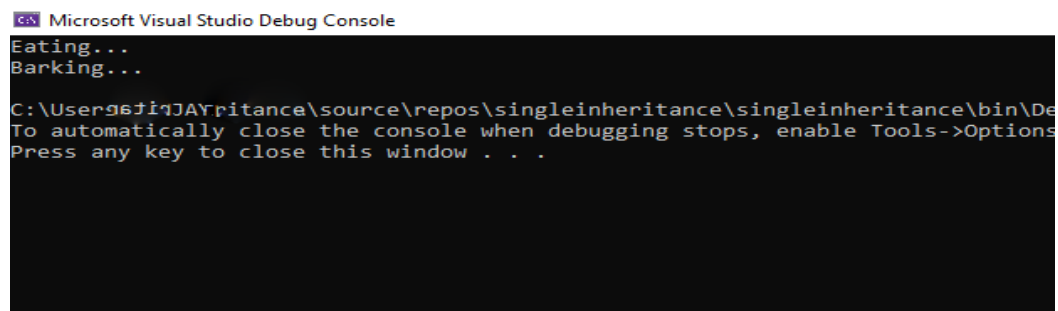
```


Task 19 Write a program for single inheritance.



```
1 using System;
2 public class Animal
3 {
4     public void eat() { Console.WriteLine("Eating..."); }
5 }
6 public class Dog : Animal
7 {
8     public void bark() { Console.WriteLine("Barking..."); }
9 }
10 class TestInheritance2
11 {
12     public static void Main(string[] args)
13     {
14         Dog d1 = new Dog();
15         d1.eat();
16         d1.bark();
17     }
18 }
```

Output:-



```
Microsoft Visual Studio Debug Console
Eating...
Barking...

C:\Users\g6j1qJAY\ritance\source\repos\singleinheritance\singleinheritance\bin\De
To automatically close the console when debugging stops, enable Tools->Options
Press any key to close this window . . .
```

Task 20 Write a program for multilevel inheritance.

```

1  using System;
2
3  namespace multilevelinheritance
4  {
5      class Son : Father
6      {
7          public void DisplayTwo()
8          {
9              Console.WriteLine("Son.. ");
10         }
11         static void Main(string[] args)
12         {
13             Son s = new Son();
14             s.Display();
15             s.DisplayOne();
16             s.DisplayTwo();
17             Console.Read();
18         }
19     }
20     class Grandfather
21     {
22         public void Display()
23         {
24             Console.WriteLine("Grandfather...");
25         }
26     }
27     class Father : Grandfather
28     {
29         public void DisplayOne()

```

```

17  Console.Read();
18  }
19  }
20  class Grandfather
21  {
22      public void Display()
23      {
24          Console.WriteLine("Grandfather...");
25      }
26  }
27  class Father : Grandfather
28  {
29      public void DisplayOne()
30      {
31          Console.WriteLine("Father...");
32      }
33  }
34  }

```

Output:-

```

Grandfather...
Father...
Son..

```

Task 21 Write a program for multiple inheritance.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace MultipleInheritApplication
7  {
8      1 reference
9      interface calc1
10     {
11         2 references
12         int add(int a, int b);
13     }
14     1 reference
15     interface calc2
16     {
17         2 references
18         int sub(int x, int y);
19     }
20     1 reference
21     interface calc3
22     {
23         2 references
24         int mul(int r, int s);
25     }
26     1 reference
27     interface calc4
28     {
29         2 references
30         int div(int c, int d);
31     }
32     2 references
33     class Calculation : calc1, calc2, calc3, calc4
34     {
35         public int result1;
36         2 references
37         public int add(int a, int b)
38         {
39             return result1 = a + b;
40         }
41     }
42 }

```

```

32     2 references
33     public int sub(int x, int y)
34     {
35         return result2 = x - y;
36     }
37     2 references
38     public int mul(int r, int s)
39     {
40         return result3 = r * s;
41     }
42     2 references
43     public int div(int c, int d)
44     {
45         return result4 = c / d;
46     }
47     0 references
48     class Program
49     {
50         0 references
51         static void Main(string[] args)
52         {
53             Calculation c = new Calculation();
54             c.add(8, 2);
55             c.sub(20, 10);
56             c.mul(5, 2);
57             c.div(20, 10);
58             Console.WriteLine("Multiple Inheritance concept Using Interfaces :\n ");
59             Console.WriteLine("Addition: " + c.result1);
60             Console.WriteLine("Substraction: " + c.result2);
61             Console.WriteLine("Multiplication: " + c.result3);
62             Console.WriteLine("Division: " + c.result4);
63             Console.ReadKey();
64         }
65     }

```

Output:-

```

Microsoft Visual Studio Debug Console
Multiple Inheritance concept Using Interfaces :

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

C:\tionrs\9DHA\e\rep\9DHA\source\repos\multipleinheritance\multipleinheritance\bin\Debug\netcoreapp3.1\multipleinheritance.exe (process)
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 22 Write a program for method overloading .

```

methodoverloading MyApplication.Program PlusMethod(int x, int y)
1 using System;
2
3 namespace MyApplication
4 {
5     0 references
6     class Program
7     {
8         1 reference
9         static int PlusMethod(int x, int y)
10        {
11            return x + y;
12        }
13
14        1 reference
15        static double PlusMethod(double x, double y)
16        {
17            return x + y;
18        }
19
20        0 references
21        static void Main(string[] args)
22        {
23            int myNum1 = PlusMethod(8, 5);
24            double myNum2 = PlusMethod(4.3, 6.26);
25            Console.WriteLine("Int: " + myNum1);
26            Console.WriteLine("Double: " + myNum2);
27        }
28    }
29 }

```

Output:-

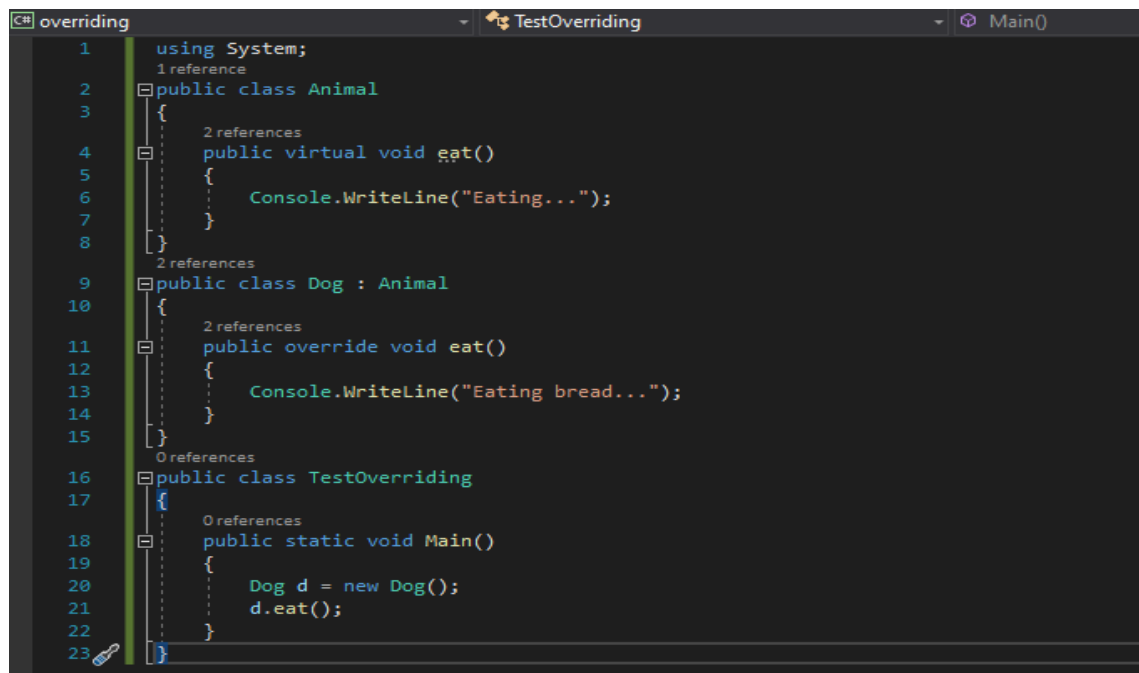
```

Microsoft Visual Studio Debug Console
Int: 13
Double: 10.559999999999999

C:\Users\obod\z\onibsource\repos\methodoverloading\methodoverloading\bin\Debug\netcoreapp3.1\methodoverloading.exe (process)
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 . . .

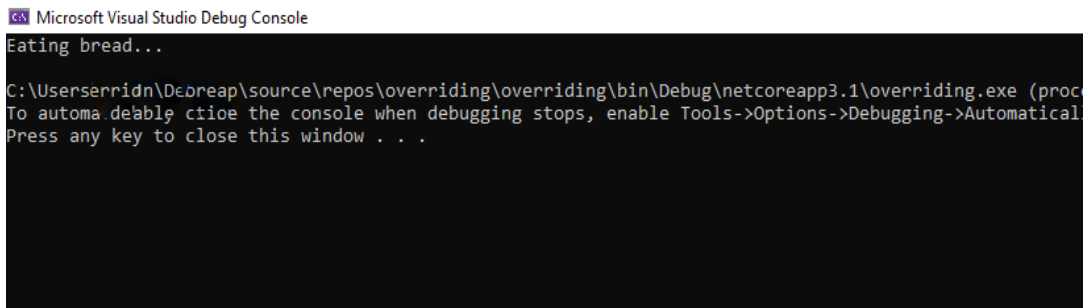
```

Task 23 Write a program for method overriding.



```
1  using System;
2  public class Animal
3  {
4      public virtual void eat()
5      {
6          Console.WriteLine("Eating...");
7      }
8  }
9  public class Dog : Animal
10 {
11     public override void eat()
12     {
13         Console.WriteLine("Eating bread...");
14     }
15 }
16 public class TestOverriding
17 {
18     public static void Main()
19     {
20         Dog d = new Dog();
21         d.eat();
22     }
23 }
```

Output:-



```
Microsoft Visual Studio Debug Console
Eating bread...
C:\Userserridn\Desktop\source\repos\overriding\overriding\bin\Debug\netcoreapp3.1\overriding.exe (process)
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatical
Press any key to close this window . . .
```

Task 24 Write a program for interface.

```

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
32     0 references
33     class Program
34     {
35         0 references
36         static void Main(string[] args)
37         {
38             DemoClass myObj = new DemoClass();
39             myObj.myMethod();
40             myObj.myOtherMethod();
41         }
42     }

```

Output:-

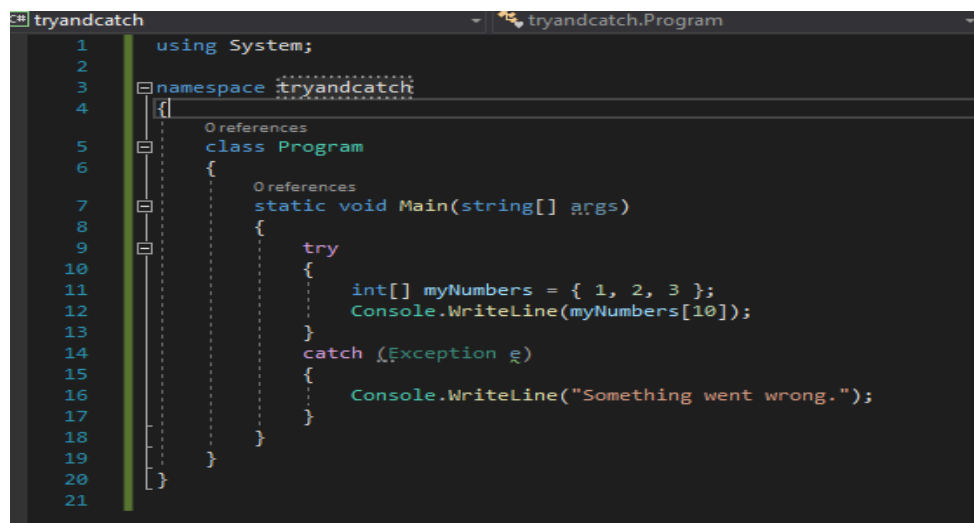
```

Microsoft Visual Studio Debug Console
Some text..
Some other text...

C:\Users\epos\source\repos\IA2\source\repos\interfaces\interfaces\bin\Debug\netcoreapp3.1\interfaces.exe (process 14668) exit
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the co
Press any key to close this window . . .

```

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

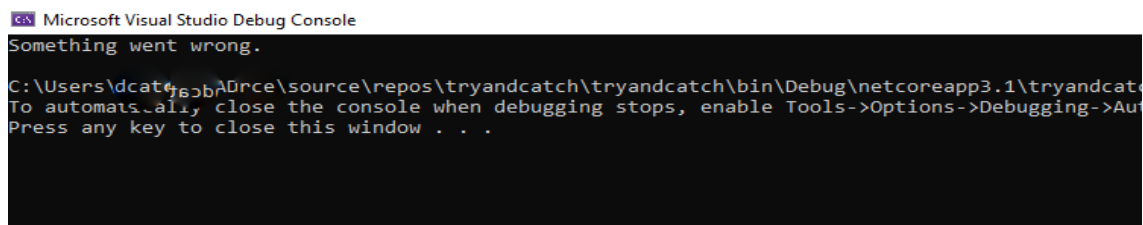


```

1  using System;
2
3  namespace tryandcatch
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              try
10             {
11                 int[] myNumbers = { 1, 2, 3 };
12                 Console.WriteLine(myNumbers[10]);
13             }
14             catch (Exception e)
15             {
16                 Console.WriteLine("Something went wrong.");
17             }
18         }
19     }
20 }
21

```

Output:-

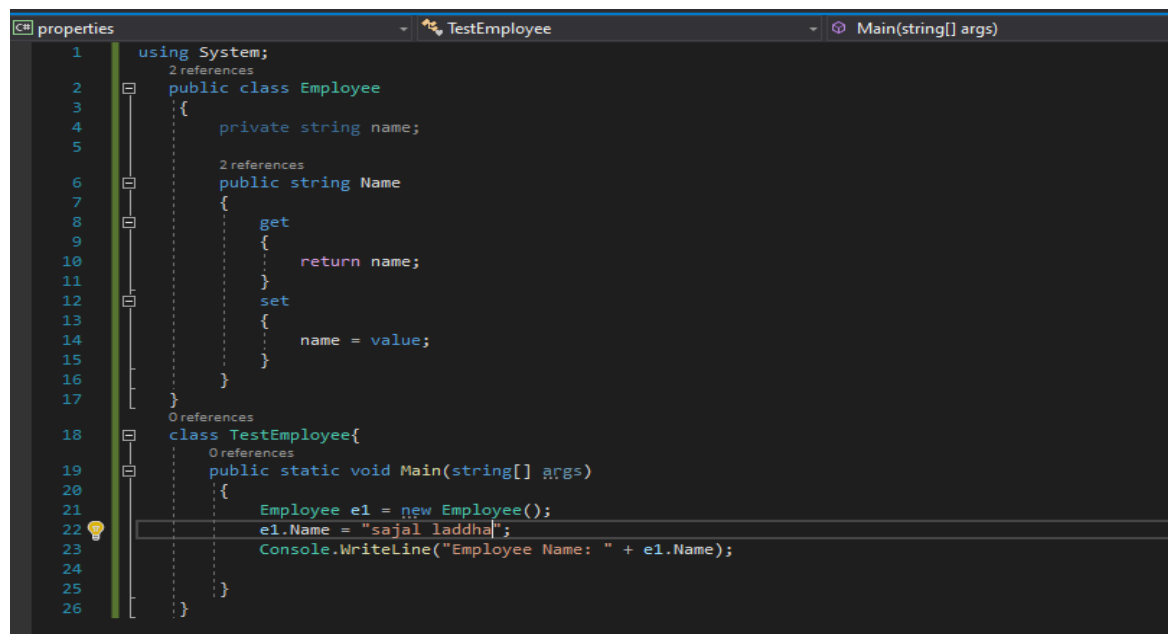


```

Microsoft Visual Studio Debug Console
Something went wrong.
C:\Users\dca...source\repos\tryandcatch\tryandcatch\bin\Debug\netcoreapp3.1\tryandcat...
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Auto...
Press any key to close this window . . .

```

Task 26 Write a program for Properties.



```

1  using System;
2  public class Employee
3  {
4      private string name;
5
6      public string Name
7      {
8          get
9          {
10             return name;
11          }
12          set
13          {
14             name = value;
15          }
16      }
17  }
18  class TestEmployee{
19      public static void Main(string[] args)
20      {
21          Employee e1 = new Employee();
22          e1.Name = "sajal laddha";
23          Console.WriteLine("Employee Name: " + e1.Name);
24      }
25  }
26

```

Output:-

```

Microsoft Visual Studio Debug Console
Employee Name: sajal laddha

C:\Users\source\AA\source\repos\properties\properties\bin\Debug\netcoreapp3.1\properties.exe (process 14764) 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 27 Write a program for Threading.

```

threading GFG mythread()
1 using System;
2 using System.Threading;
3
4 class GFG
5 {
6
7     // Main method
8     static void Main(string[] args)
9     {
10
11         // Creating and initializing thread
12         Thread thr = new Thread(mythread);
13         thr.Start();
14         Console.WriteLine("Main Thread Ends!!");
15     }
16
17     // Static method
18     static void mythread()
19     {
20         for (int c = 0; c <= 3; c++)
21         {
22             Console.WriteLine("mythread is in progress!!");
23             Thread.Sleep(1000);
24         }
25         Console.WriteLine("mythread ends!!");
26     }
27 }
28

```

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!!

C:\Users\source\AA\source\repos\threading\threading\bin\Debug\netcoreapp3.1\threading.exe (process 14992) 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 28 Write a program to access data from database using ADO.NET

```

1  using System;
2  using System.Windows.Forms;
3  using System.Data.SqlClient;
4  namespace WindowsApplication1
5  {
6      1 reference
7      public partial class Form1 : Form
8      {
9          0 references
10         public Form1()
11         {
12             InitializeComponent();
13         }
14         0 references
15         private void button1_Click(object sender, EventArgs e)
16         {
17             string connetionString = null;
18             SqlConnection connection;
19             SqlCommand command;
20             string sql = null;
21             SqlDataReader dataReader;
22             connetionString = "Data Source=ServerName;Initial Catalog=DatabaseName;User ID=UserName;Passwo
23             sql = "Your SQL Statement Here , like Select * from product";
24             connection = new SqlConnection(connetionString);
25             try
26             {
27                 connection.Open();
28                 command = new SqlCommand(sql, connection);
29                 dataReader = command.ExecuteReader();
30                 while (dataReader.Read())
31                 {
32                     MessageBox.Show(dataReader.GetValue(0) + " - " + dataReader.GetValue(1) + " - " + data
33                 }
34                 dataReader.Close();
35                 command.Dispose();
36                 connection.Close();
37             }
38         }
39     }
40 }

```

```

10  0 references
11  private void button1_Click(object sender, EventArgs e)
12  {
13      string sql = null;
14      SqlDataReader dataReader;
15      connetionString = "Data Source=ServerName;Initial Catalog=DatabaseName;User ID=UserName;Passwo
16      sql = "Your SQL Statement Here , like Select * from product";
17      connection = new SqlConnection(connetionString);
18      try
19      {
20          connection.Open();
21          command = new SqlCommand(sql, connection);
22          dataReader = command.ExecuteReader();
23          while (dataReader.Read())
24          {
25              MessageBox.Show(dataReader.GetValue(0) + " - " + dataReader.GetValue(1) + " - " + dat
26          }
27          dataReader.Close();
28          command.Dispose();
29          connection.Close();
30      }
31      catch (Exception ex)
32      {
33          MessageBox.Show("Can not open connection ! ");
34      }
35  }
36  }
37  }
38  }
39  }
40  }
41  }

```

Task 29 Write a programme using namespace.

```

1  using System;
2
3
4  namespace first_space
5  {
6      2 references
7      class namespace_cl
8      {
9          1 reference
10         public void func()
11         {
12             Console.WriteLine("Inside first_space");
13         }
14     }
15
16     namespace second_space
17     {
18         2 references
19         class namespace_cl
20         {
21             1 reference
22             public void func()
23             {
24                 Console.WriteLine("Inside second_space");
25             }
26         }
27     }
28
29     0 references
30     class TestClass
31     {
32         0 references
33         static void Main(string[] args)
34         {
35             first_space.namespace_cl fc = new first_space.namespace_cl();
36             second_space.namespace_cl sc = new second_space.namespace_cl();
37             fc.func();
38             sc.func();
39             Console.ReadKey();
40         }
41     }
42 }

```

```

14 namespace second_space
15 {
16     2 references
17     class namespace_cl
18     {
19         1 reference
20         public void func()
21         {
22             Console.WriteLine("Inside second_space");
23         }
24     }
25
26     0 references
27     class TestClass
28     {
29         0 references
30         static void Main(string[] args)
31         {
32             first_space.namespace_cl fc = new first_space.namespace_cl();
33             second_space.namespace_cl sc = new second_space.namespace_cl();
34             fc.func();
35             sc.func();
36             Console.ReadKey();
37         }
38     }
39 }

```

Output:-

```

Microsoft Visual Studio Debug Console
Inside first_space
Inside second_space
C:\User29m6N\JAY\1\9271A\source\repos\namespace\namespace\bin\Debug\netcoreapp3.1\namespace.exe (process 13040) exited wi
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the cons
Press any key to close this window . . .

```

