DEV SANSKRITI VISHWAVIDYALAYA

Practical File

Subject: C# (sharp)

Submitted to:

Mr. Chandrashekhar Patel

Submitted by:

Amisha Singh

BCA (Vth sem)

INDEX

1) Write a program to print Armstrong numbers.

```
using System;
  public class ArmstrongExample
   {
     public static void Main(string[] args)
      {
       int  n,r,sum=0,temp;
       Console.Write("Enter the Number= ");
       n= int.Parse(Console.ReadLine());
       temp=n;
       while(n>0)
       {
        r=n%10;
        sum=sum+(r*r*r);
        n=n/10;
       }
       if(temp==sum)
        Console.Write("Armstrong Number.");
       else
       Console.Write("Not Armstrong Number.");
      }
  }
```

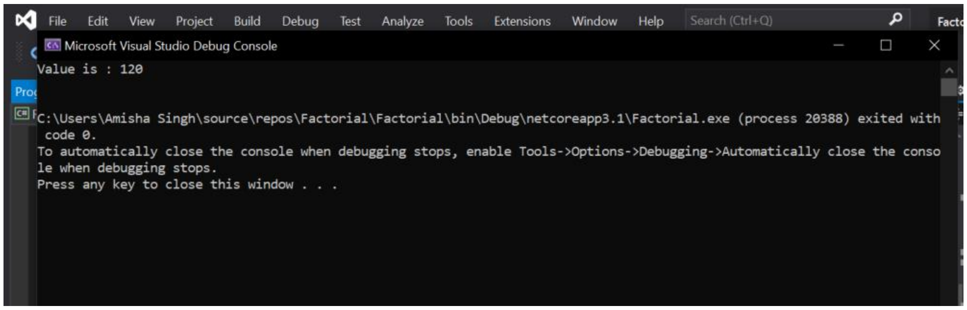
OUTPUT:

[redacted]

2) Write a program to print factorial of a number.

```
using System;

namespace MyApplication

 {

   class Factorial

 {

      public int display(int n)

 {

         int res = 1;

        while (n != 1)

 {

            res = res * n;

            n = n - 1;

        }

     return res;

      }

    static void Main(string[] args)

    {

       int value = 5;

       int ret;

         Factorial fact = new Factorial();

         ret = fact.display(value);

         Console.WriteLine("Value is : {0}", ret );

         Console.ReadLine();
} } }
```

OUTPUT:

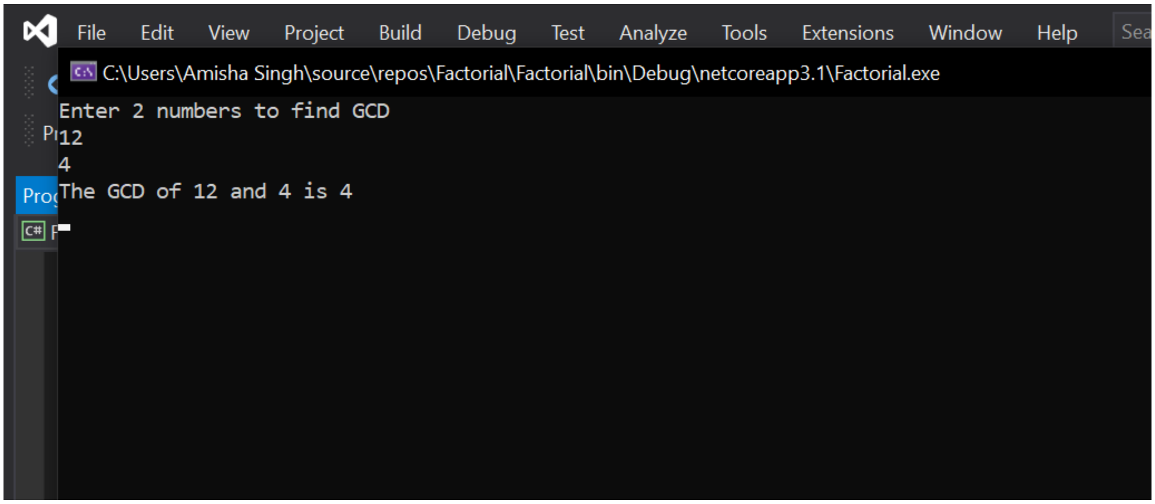


```
Value is : 120

C:\Users\Amisha Singh\source\repos\Factorial\Factorial\bin\Debug\netcoreapp3.1\Factorial.exe (process 20388) exited with
 code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

3) Write a program to find GCD of two numbers.

```
using System;

class myclass
{
   static void Main()
   {
      int i1, i2;

      Console.WriteLine("Enter 2 numbers to find GCD");
      i1 = int.Parse(Console.ReadLine());
      i2 = int.Parse(Console.ReadLine());

      int n1, n2;
      //Making sure n1 is greater than n2
      if (i1 > i2)
      {
         n1 = i1;
         n2 = i2;
      }
      else
      {
         n1 = i2;
         n2 = i1;
      }
      int result = gcd(n1, n2);
      Console.WriteLine("The GCD of {0} and {1} is {2}",i1,i2,result);
      Console.Read();
   }

   private static int gcd(int n1, int n2)
   {
      int rem = 5;
      while (n2 > 0)
      {
         rem = n1 % n2;
         if (rem == 0)
            return n2;
         n1 = n2;
         n2 = rem;

      }
      //gcd of any number with 0 is number itself. 06

      return n1;

   }
}
```

OUTPUT:


File
Edit
View
Project
Build
Debug
Test
Analyze
Tools
Extensions
Window
Help
Sea
C:\Users\Amisha Singh\source\repos\Factorial\Factorial\bin\Debug\netcoreapp3.1\Factorial.exe

```
Enter 2 numbers to find GCD
12
4
The GCD of 12 and 4 is 4
```

4) Write a program to find if the number is prime or not.

```
using System;
  public class PrimeNumberExample
   {
     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:

5) Write a program to print fibonacci series.

```
using System;
  public class FibonacciExample
   {
     public static void Main(string[] args)
      {
         int n1=0,n2=1,n3,i,number;
         Console.Write("Enter the number of elements: ");
         number = int.Parse(Console.ReadLine());
         Console.Write(n1+" "+n2+" "); //printing 0 and 1
         for(i=2;i<number;++i) //loop starts from 2 because 0 and 1 are already printed
         {
          n3=n1+n2;
          Console.Write(n3+" ");
          n1=n2;
          n2=n3;
         }
      }
   }
```

OUTPUT:

```
    }
```

6) Write a program to print half pyramid pattern.

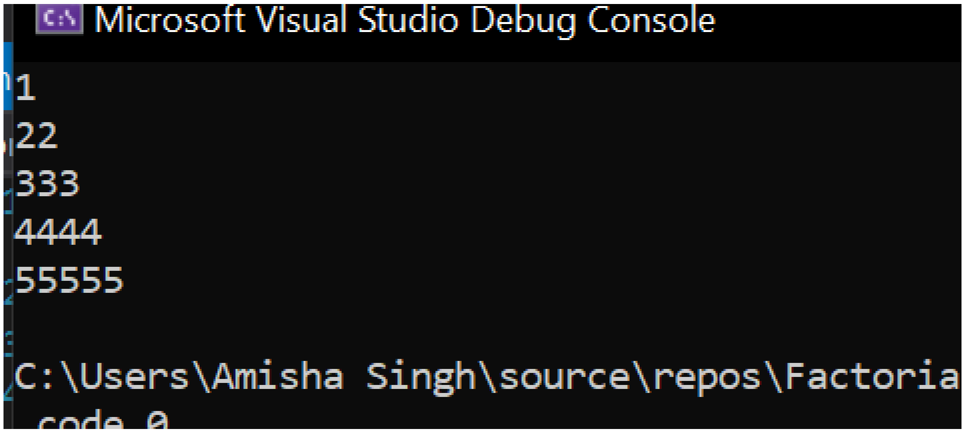
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int count = 6;
            string print = "*";
            string space = " ";
            for (int i = 0; i < count; i++)
            {
                for (int j = count - 1; j > i; j--)
                {
                    Console.Write(space);
                }
                Console.WriteLine(print);
                print = print + "*";
            }
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

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

```
class pattern
{
public static void main()
{
int num=5;
for(int=1;I<=num;I++)
{
for(int j=1;j<=I;j++)
{
Console.Write(i);
}
Console.WriteLine();
}
Console.ReadKey();
}
}
```

OUTPUT:

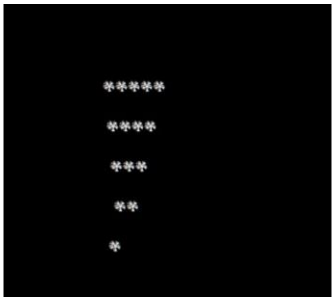


```
1
22
333
4444
55555

C:\Users\Amisha Singh\source\repos\Factoria
 code 0
```

8) Write a program to print half pyramid inverse.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication1

{

    class Program

    {

        static void Main(string[] args)

        {

            int count = 6;

            string print = "*";

            string space = " ";

            for (int i = 0; i > count; I--)

            {

                for (int j = 1; j <=i; j++)

                {

                    Console.Write(print);

                }

                Console.WriteLine(space);

                print = print + "*";

            }

            Console.ReadLine();

        }

    }
}
```

OUTPUT:

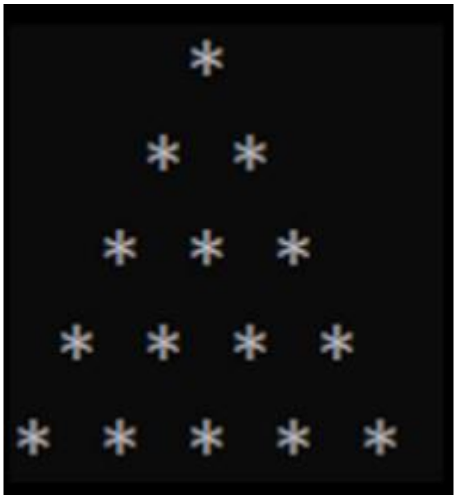


```
*****
****
***
**
*
```

9) Write a program to print pyramid pattern.

```
using System;
namespace SampleCode
{
    class Program
      {
         static void Main(string[] args)
          {
             int x = 15; // Total Number of Lines...
             for (int i = 1; i<= x; i++)
              {
                 //loop to print spaces
                 for (int j = 1; j <= (x - i); j++)
                    Console.Write(" ");

                 //loop to print stars
                for (int t = 1; t <i * 2; t++)
                   Console.Write("*");
                Console.WriteLine();
              }
           Console.ReadLine();
        }
    }
}
```

OUTPUT:

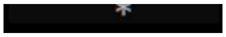


```
    *
   * *
  * * *
 * * * *
* * * * *
```

10) Write a program to print inverse pyramid pattern.

```
using System;

class Stars

{

public static void Main()

{

int i, j, k;

int l = 5;

for (i = 0; i < 5; i++)

{

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

{

  Console.Write(" ");

   }

for (k = 0; k < l; k++)

{

Console.Write("*");

}

l--;

Console.WriteLine();

}

}

}
```

OUTPUT:

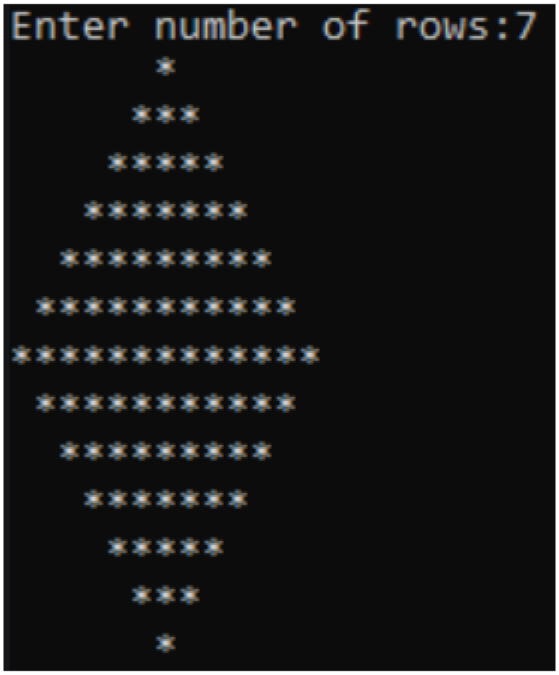


```
*
```

11) Write a program to print diamond pattern.

```
 private static void DiamondOne()
   {
     int i, j, count = 1, number;
     Console.Write("Enter number of rows:");
     number = int.Parse(Console.ReadLine());
     count = number - 1;
     for (j = 1; j <= number; j++)
     {
        for (i = 1; i <= count; i++)
           Console.Write(" ");
        count--;
        for (i = 1; i <= 2 * j - 1; i++)
         Console.Write("*");
         Console.WriteLine();
      }
      count = 1;
     for (j = 1; j <= number - 1; j++)
      {
        for (i = 1; i <= count; i++)
            Console.Write(" ");
         count++;
         for (i = 1; i <= 2 * (number - j) - 1; i++)
            Console.Write("*");
         Console.WriteLine();
      }
      Console.ReadLine();
}
```

OUTPUT:

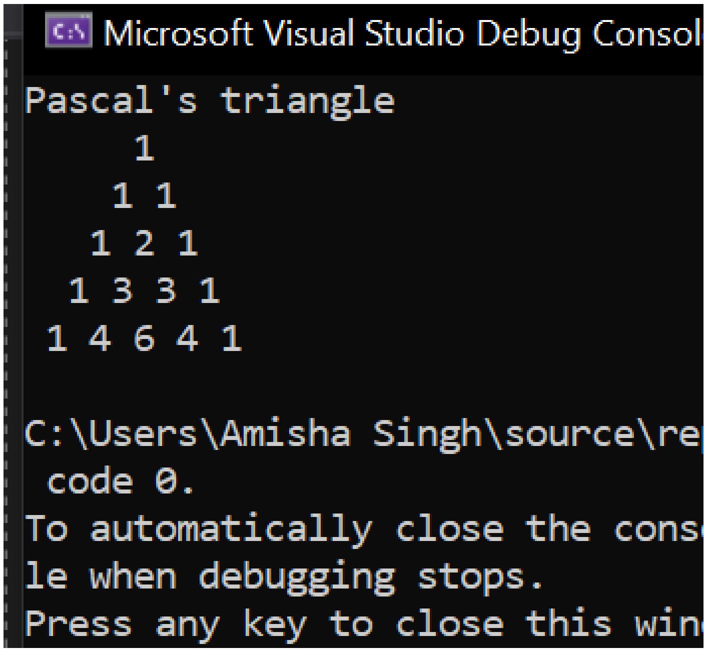


```
Enter number of rows:7
      *
     ***
    *****
   *******
  *********
 ***********
*************
 ***********
  *********
   *******
    *****
     ***
      *
```

12) Write a program to print pascal’s triangle.

```
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:

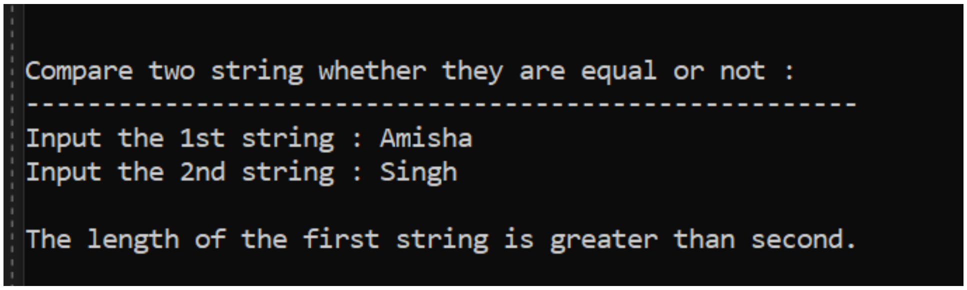


```
Pascal's triangle
     1
    1 1
   1 2 1
  1 3 3 1
 1 4 6 4 1

C:\Users\Amisha Singh\source\re
 code 0.
To automatically close the cons
le when debugging stops.
Press any key to close this win
```

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

```
using System;
 public class Exercise6
 {
public static void Main()
 {
 string str1, str2;
int flg=0;
int i=0,l1,l2,yn=0;
 Console.Write("\n\nCompare two string whether they are equal or not :\n");
 Console.Write("-----------------------------------------------\n");
 Console.Write("Input the 1st string : ");
 str1 = Console.ReadLine();
 Console.Write("Input the 2nd string : ");
 str2 = Console.ReadLine();
 l1=str1.Length;
 l2=str2.Length;
 if(l1==l2)
 {
 for(i=0;i<l1;i++)
{
 if(str1[i] != str2[i])
 {
 yn=1; i=l1;
}
}
}
 if(l1 == l2)
flg=0;
else if(l1 > l2)
 flg=1;
else if(l1 < l2)
 flg=-1;
 if(flg == 0)
{
 if(yn==0)
Console.Write("\nThe length of both strings are equal and \nalso, both strings are same.\n\n");
else
 Console.Write("\nThe length of both strings are equal \nbut they are not same.\n\n");
 }
 else if(flg == -1)
{
Console.Write("\nThe length of the first string is smaller than second.\n\n");
 }
 else
{
 Console.Write("\nThe length of the first string is greater than second.\n\n");
 }
}
}
```

OUTPUT:

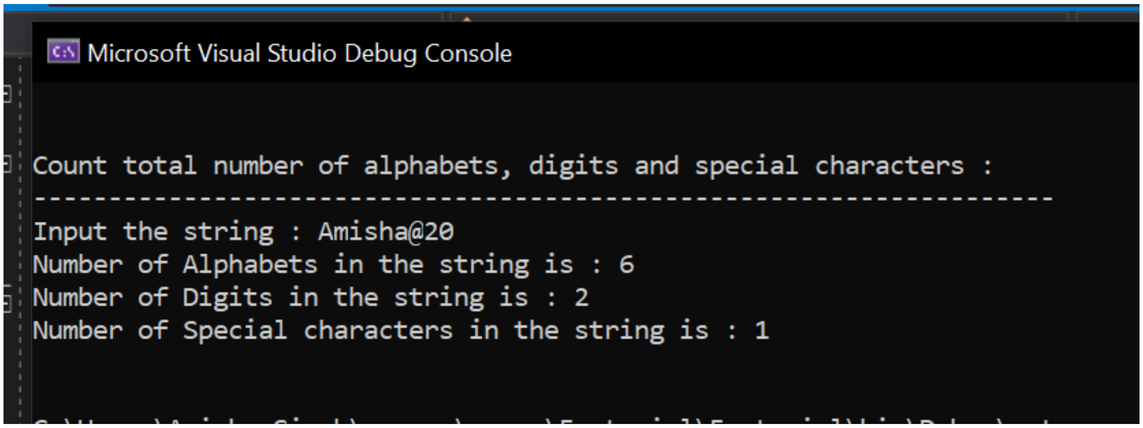


```

Compare two string whether they are equal or not :
------------------------------------------------------
Input the 1st string : Amisha
Input the 2nd string : Singh

The length of the first string is greater than second.
```

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

```
using System;
public class Exercise7
{
public static void Main()
{
 string str;
int alp, digit, splch, i,l;
alp = digit = splch = i = 0;
Console.Write("\n\nCount total number of alphabets, digits and special characters :\n");
Console.Write("------------------------------------------------------------\n");
Console.Write("Input the string : ");
str = Console.ReadLine();
l=str.Length; /* Checks each character of string*/
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:

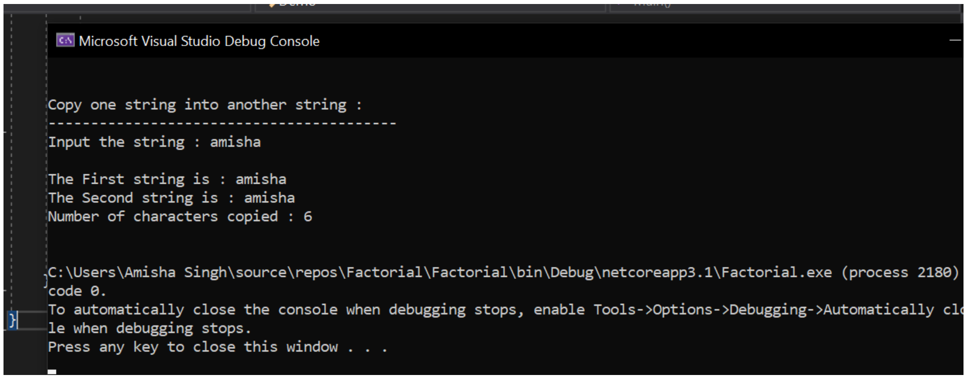


```
Count total number of alphabets, digits and special characters :
---------------------------------------------------------------------
Input the string : Amisha@20
Number of Alphabets in the string is : 6
Number of Digits in the string is : 2
Number of Special characters in the string is : 1
```

15) Write a program to copy one string into another string.

```
using System;
public class Demo
 {
 public static void Main()
 {
 string str1;
 int i,l;
 Console.Write("\n\nCopy one string into another string :\n");
 Console.Write("----------------------------------------\n");
 Console.Write("Input the string : ");
 str1 = Console.ReadLine();
 l=str1.Length;
 string[] str2=new string[l];
i=0;
 while(i<l)
 {
 string tmp=str1[i].ToString();
 str2[i] = tmp; i++;
}
Console.Write("\nThe First string is : {0}\n", str1);
Console.Write("The Second string is : {0}\n", string.Join("",str2));
Console.Write("Number of characters copied : {0}\n\n", i);
 }
 }
```

OUTPUT:

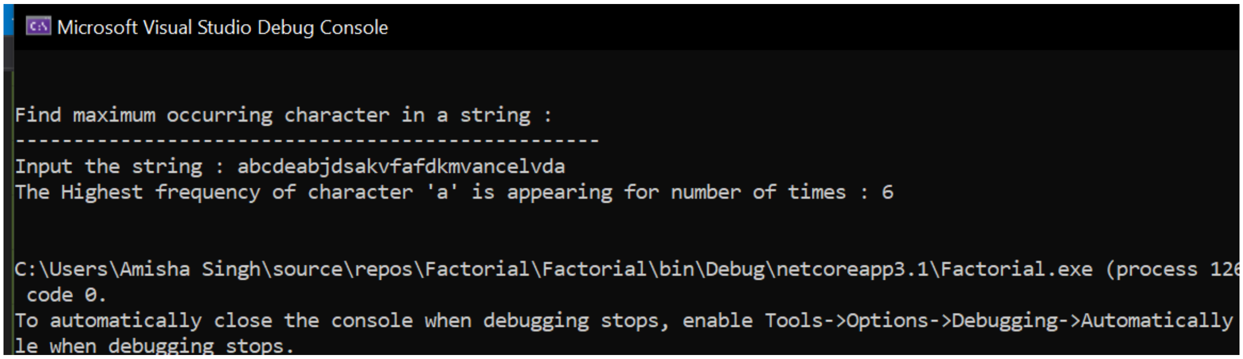


```
Copy one string into another string :
----------------------------------------
Input the string : amisha

The First string is : amisha
The Second string is : amisha
Number of characters copied : 6


C:\Users\Amisha Singh\source\repos\Factorial\Factorial\bin\Debug\netcoreapp3.1\Factorial.exe (process 2180)
code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically clo
le when debugging stops.
Press any key to close this window . . .
```

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

```
using System;

public class Exercise10

{

 public static void Main()

 {

 string str;

int[] ch_fre = new int[255];

int i = 0, max,l;

int ascii;

Console.Write("\n\nFind maximum occurring character in a string :\n"); Console.Write("----------------
--------------------------------\n"); Console.Write("Input the string : "); str = Console.ReadLine();
l=str.Length;

for(i=0; i<255; i++) //Set frequency of all characters to 0

{

ch_fre[i] = 0;

} /* Read for frequency of each characters */

i=0;

while(i<l)

{

ascii = (int)str[i];

ch_fre[ascii] += 1;

i++;

} // Console.Write("{0} ",(char)65);

max = 0;

for(i=0; i<255; i++)

{

if(i!=32)

{

if(ch_fre[i] > ch_fre[max]) max = i;

}

}
Console.Write("The Highest frequency of character '{0}' is appearing for number of times : {1} \n\n",
(char)max, ch_fre[max]);

}

}
```

OUTPUT:

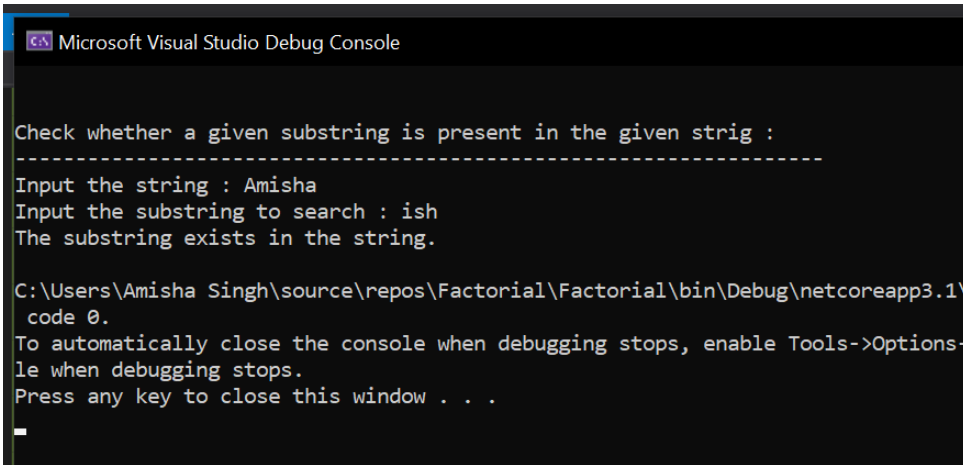


```


Find maximum occurring character in a string :
--------------------------------------------------
Input the string : abcdeabjdsakvfafdkmvancelvda
The Highest frequency of character 'a' is appearing for number of times : 6


C:\Users\Amisha Singh\source\repos\Factorial\Factorial\bin\Debug\netcoreapp3.1\Factorial.exe (process 126
 code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically
le when debugging stops.
```

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

```
using System;

public class Program

{

public static void Main()

{

 string str1,str2;

 bool m;

 Console.Write("\n\nCheck whether a given substring is present in the given strig :\n");

    Console.Write("-----------------------------------------------------------------\n");

       Console.Write("Input the string : ");

         str1 = Console.ReadLine();

   Console.Write("Input the substring to  search : ");

         str2 = Console.ReadLine();

            m=str1.Contains(str2);

Console.Write("The substring exists in the string.\n\n");

}

}
```

OUTPUT:

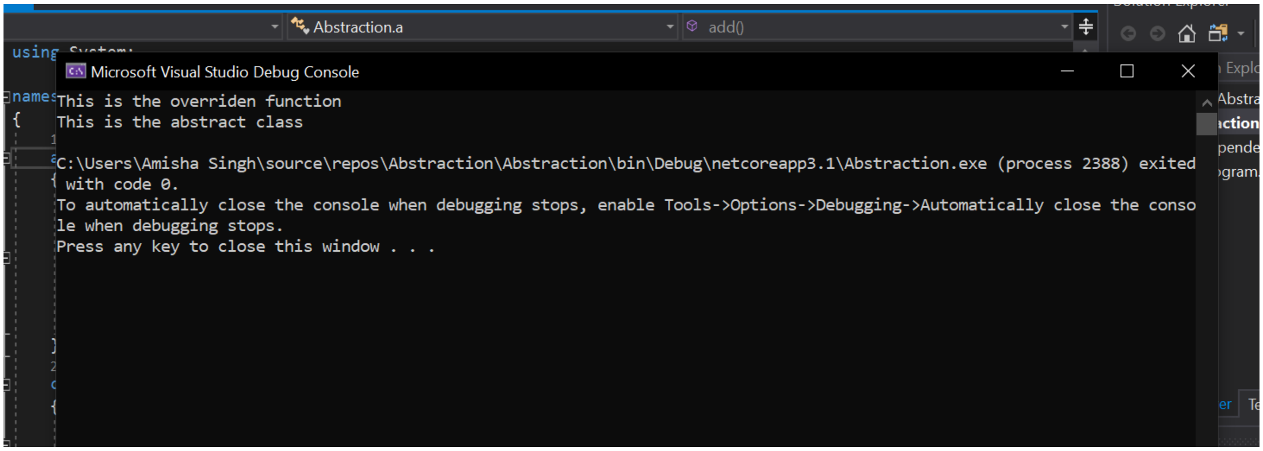


```


Check whether a given substring is present in the given strig :
-------------------------------------------------------------------
Input the string : Amisha
Input the substring to search : ish
The substring exists in the string.

C:\Users\Amisha Singh\source\repos\Factorial\Factorial\bin\Debug\netcoreapp3.1\
 code 0.
To automatically close the console when debugging stops, enable Tools->Options-
le when debugging stops.
Press any key to close this window . . .
```

18) Write a program for Abstraction.

```
using System;


namespace Abstraction
{
    abstract class a
    {
        public abstract void add();
        public void func()
        {
            Console.WriteLine("This is the abstract class");
        }
    }
    class b:a
    {
        public override void add()
        {
            Console.WriteLine("This is the overriden function");
        }
    }
    class Program
    {
        public static void Main()
        {
            b obj = new b();
            obj.add();
            obj.func();
        }
    }
}
```

OUTPUT:

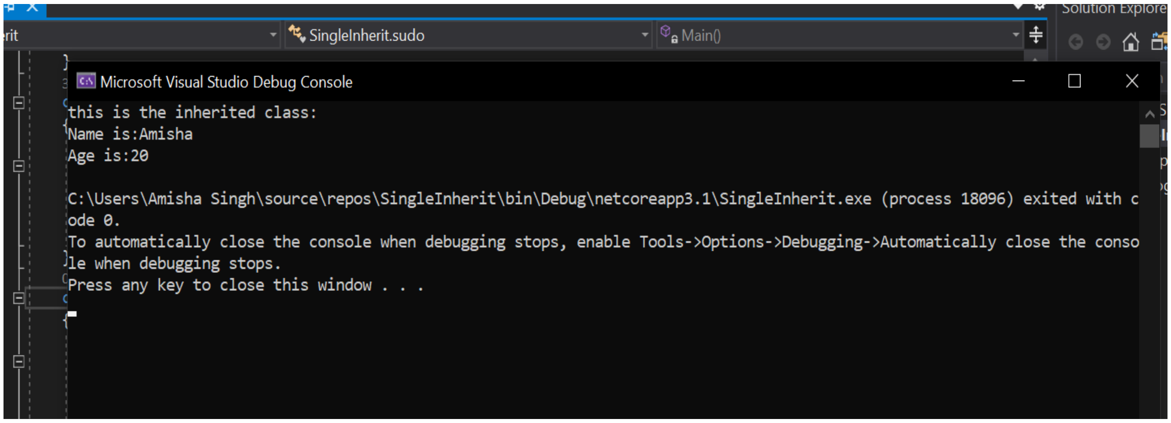


```
This is the overriden function
This is the abstract class

C:\Users\Amisha Singh\source\repos\Abstraction\Abstraction\bin\Debug\netcoreapp3.1\Abstraction.exe (process 2388) exited
 with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

19) Write a program for single inheritance.

```
using System;
using System.Runtime.CompilerServices;
using System.Xml.Linq;
namespace SingleInherit
{
    class parent
    {
        public string name;
        public int age;
        public  void func(string name, int age)
        {
            this.name = name;
            this.age = age;
            Console.WriteLine("Name is:" + name);
            Console.WriteLine("Age is:" + age);
       }


    }
    class inherited : parent
    {
        public inherited()
        {
            Console.WriteLine("this is the inherited class:");
        }
    }
    class sudo
    {
       static void Main()
        {
            inherited obj = new inherited();
            obj.func("Amisha", 20);
        }
    }
}
```

OUTPUT:

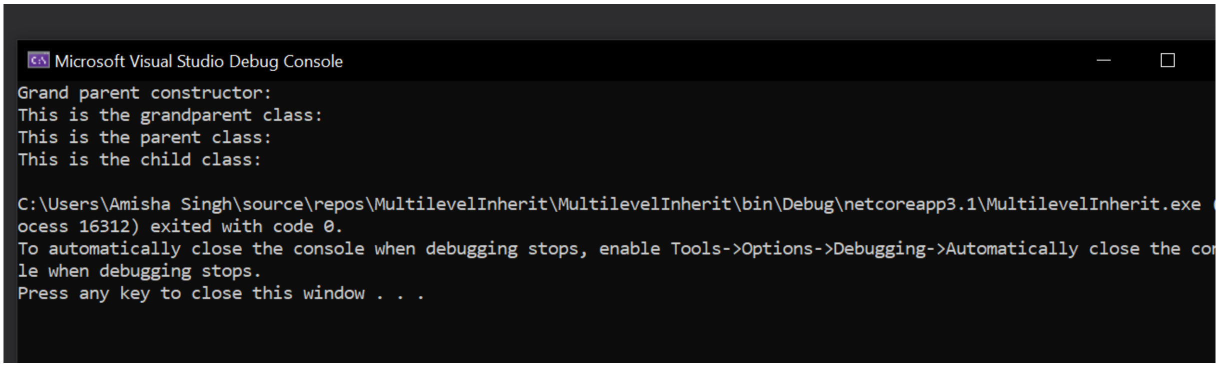


```
this is the inherited class:
Name is:Amisha
Age is:20

C:\Users\Amisha Singh\source\repos\SingleInherit\bin\Debug\netcoreapp3.1\SingleInherit.exe (process 18096) exited with c
ode 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

20) Write a program for multilevel inheritance.

```
using System;


namespace MultilevelInherit
{
    class grandparent
    {
        public grandparent()
        {
            Console.WriteLine("Grand parent constructor:");
        }
        public void func()
        {
            Console.WriteLine("This is the grandparent class:");
        }
    }
    class parent:grandparent
    {
        public void func2()
        {
            Console.WriteLine("This is the parent class:");
        }
    }
    class child : parent
    {
        public void func3()
        {
            Console.WriteLine("This is the child class:");
        }
    }
class program
    {
        public static void Main()
        {
            child cd = new child();
            cd.func();
            cd.func2();
            cd.func3();
        }
    }
}
```

OUTPUT:

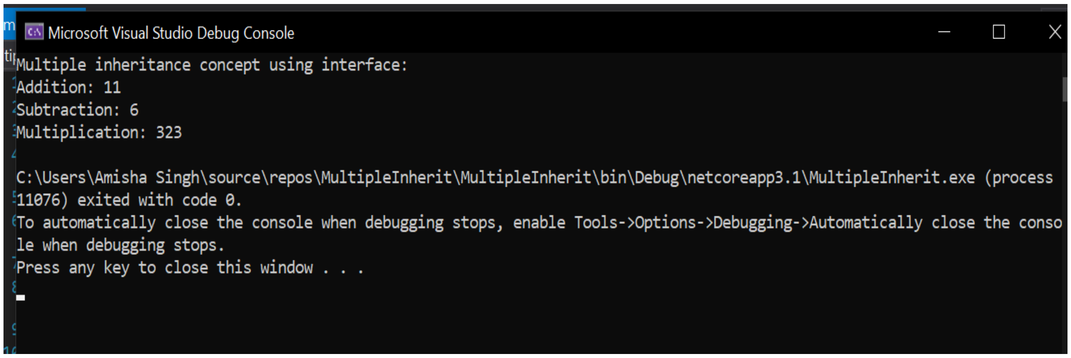


```
Grand parent constructor:
This is the grandparent class:
This is the parent class:
This is the child class:

C:\Users\Amisha Singh\source\repos\MultilevelInherit\MultilevelInherit\bin\Debug\netcoreapp3.1\MultilevelInherit.exe (
ocess 16312) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the co
le when debugging stops.
Press any key to close this window . . .
```

21) Write a program for multiple inheritance.

```
using System;
namespace MultipleInherit
{
    public interface one
    {
      int add(int a, int b);
    }
     public interface two
    {
        int sub(int x, int y);
    }
    public interface three
    {
        int mul(int n, int m);
    }
    class calc:one,two,three
    {
        public int r1;
        public int add(int a,int b)
        {
            return r1=(a + b);
        }
        public int r2;
        public int sub(int x,int y)
        {
            return r2= (x - y);
        }
        public int r3;
       public int mul(int n,int m)
{
         return r3= (n * m);
      }
   }
   class program
   {
      public static void Main()
      {
         calc obj = new calc();
         obj.add(4, 7);
         obj.sub(10, 4);
         obj.mul(19, 17);
         Console.WriteLine("Multiple inheritance concept using interface:");
         Console.WriteLine("Addition: " + obj.r1);
         Console.WriteLine("Subtraction: " + obj.r2);
         Console.WriteLine("Multiplication: " + obj.r3);
      }
   }

}
```

OUTPUT:

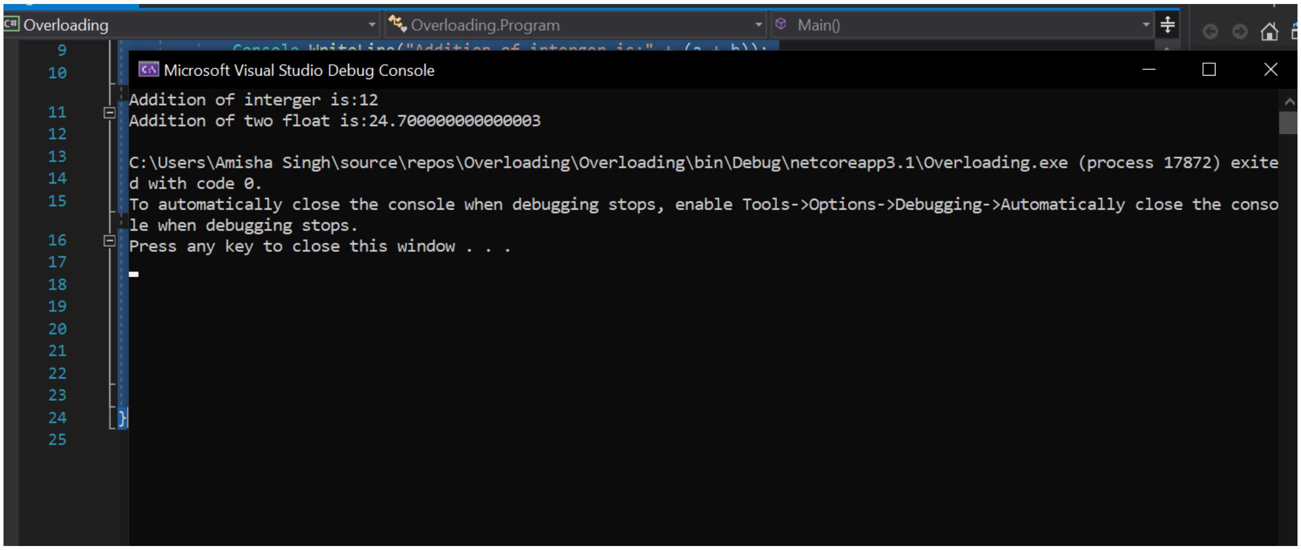


```
Multiple inheritance concept using interface:
Addition: 11
Subtraction: 6
Multiplication: 323

C:\Users\Amisha Singh\source\repos\MultipleInherit\MultipleInherit\bin\Debug\netcoreapp3.1\MultipleInherit.exe (process
11076) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

22) Write a program for method overloading.

```
using System;
namespace Overloading
{
    class Program
    {
        static void Func(int a,int b)
        {
            Console.WriteLine("Addition of interger is:" + (a + b));
        }
        static void Func(double a,double b)
        {
            Console.WriteLine("Addition of two float is:" + (a + b));


        }
        public static void Main()
        {
            Program obj = new Program();


            Func(4, 8);
            Func(4.9,19.8);
        }
    }
}
```

OUTPUT:

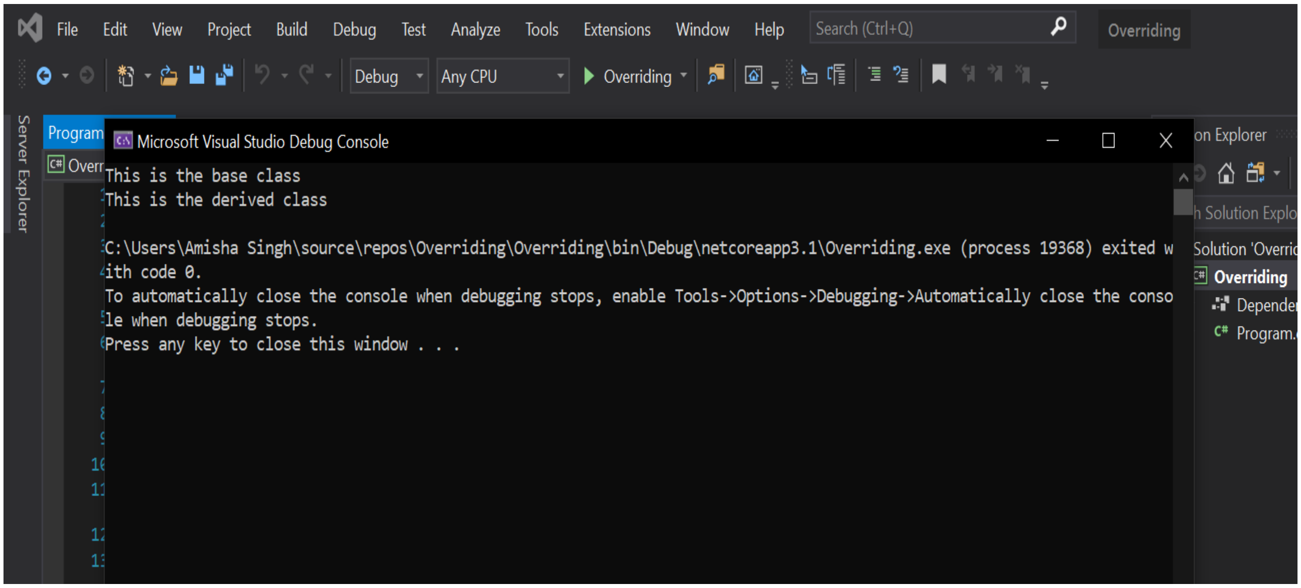


```
Addition of interger is:12
Addition of two float is:24.700000000000003

C:\Users\Amisha Singh\source\repos\Overloading\Overloading\bin\Debug\netcoreapp3.1\Overloading.exe (process 17872) exite
d with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

23) Write a program for method overriding.

```
using System;

namespace Overriding

{

    class baseclass

    {

        public virtual void show()

        {

            Console.WriteLine("This is the base class");

        }

    }

    class derived: baseclass

    {

        public override void show()

        {

            Console.WriteLine("This is the derived class");

        }

    }

    class Program

    {

        public static void Main()

        {

            baseclass obj;

            obj = new baseclass();

            obj.show();

            obj = new derived();

            obj.show();

        }

    }

}
```

OUTPUT:

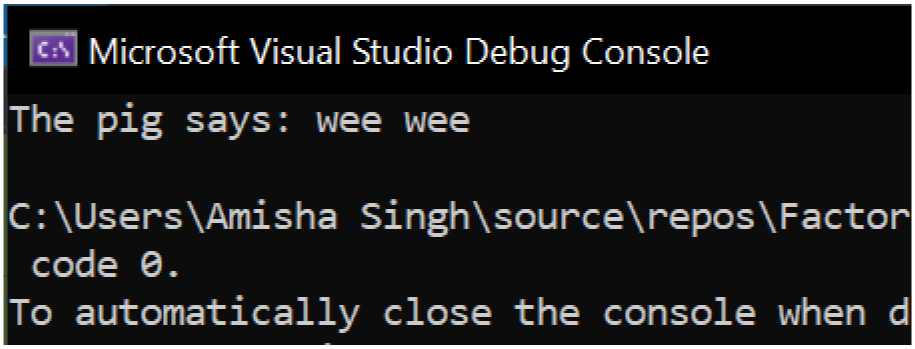


```
This is the base class
This is the derived class

C:\Users\Amisha Singh\source\repos\Overriding\Overriding\bin\Debug\netcoreapp3.1\Overriding.exe (process 19368) exited w
ith code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

24) Write a program for interface.

```
using System;
interface Animal
 {
 void animalSound();
 class Pig : Animal
 {
public void animalSound()
{
Console.WriteLine("The pig says: wee wee");
 }
 class Program {
static void Main(string[] args)
{
 Pig myPig = new Pig(); // Create a Pig object
 myPig.animalSound();
}
}
}
```

OUTPUT:

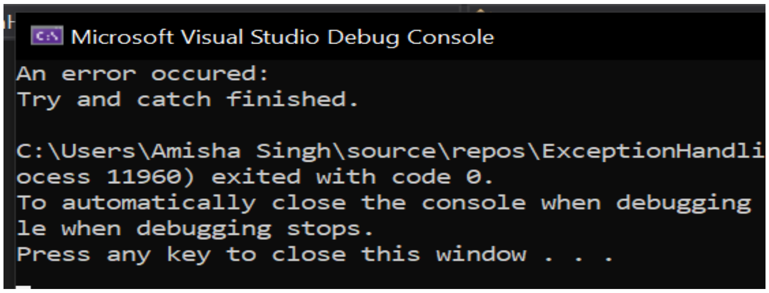


```
The pig says: wee wee

C:\Users\Amisha Singh\source\repos\Factor
 code 0.
To automatically close the console when d
```

Write a program for exception handling through try and catch.

```
using System;
 namespace ExceptionHandling
{
    class Program
    {
        public static void Main()
        {
            int[] array= { 1, 2, 3, 4, 5, 6 };
            try
            {
                Console.WriteLine(array[10]);
            }
            catch(Exception e)
            {
                Console.WriteLine("An error occured:");
            }
            finally
            {
                Console.WriteLine("Try and catch finished.");
            }
        }
    }
}
```

OUTPUT:



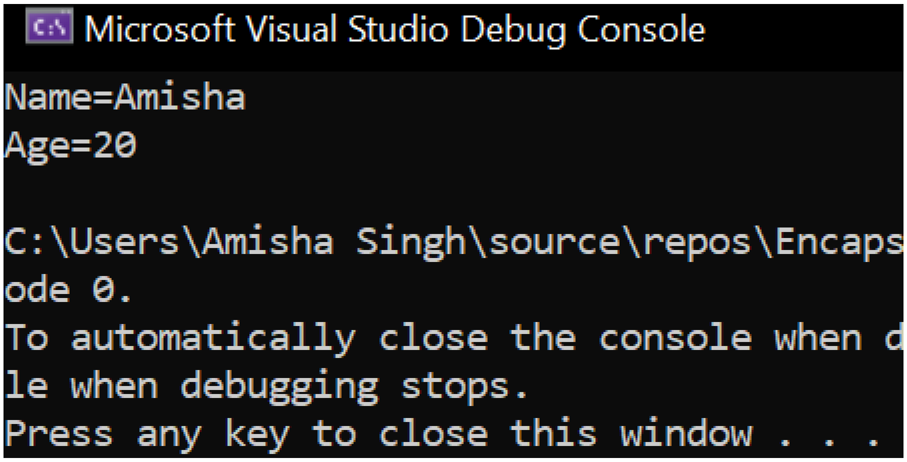
```
An error occured:
Try and catch finished.

C:\Users\Amisha Singh\source\repos\ExceptionHandli
ocess 11960) exited with code 0.
To automatically close the console when debugging
le when debugging stops.
Press any key to close this window . . .
```

26)

Write a program for Properties.

```
using System;

namespace Encapsulation
{
    public class EncapDemo
    {
        private string name;
            private int age;
        public string Name
        {
            get
            {
                return name;
            }
            set
            {
                name = value;
            }
        }
            public int Age
        {
            get
            {
                return age;
            }
            set
            {
                age = value;
            }
}

    class ABC
    {
        public static void Main()
        {
            EncapDemo obj= new EncapDemo();
            obj.Name = "Amisha";
            obj.Age = 20;
            Console.WriteLine("Name=" + obj.Name);
            Console.WriteLine("Age=" + obj.Age);
        }
    }
    }
  }
```

OUTPUT:

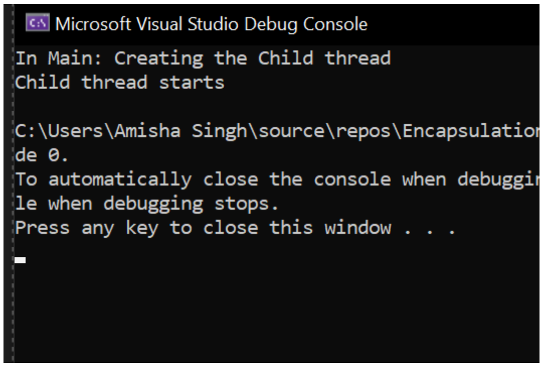


```
Name=Amisha
Age=20

C:\Users\Amisha Singh\source\repos\Encaps
ode 0.
To automatically close the console when d
le when debugging stops.
Press any key to close this window . . .
```

27) Write a program for Threading.

```
using System;

using System.Threading;

namespace MultithreadingApplication

{

class ThreadCreationProgram

{

      public static void CallToChildThread()

{

         Console.WriteLine("Child thread starts");

 }

  static void Main(string[] args)

{

     ThreadStart childref = new ThreadStart(CallToChildThread);             Console.WriteLine("In Main:
Creating the Child thread");

   Thread childThread = new Thread(childref);

      childThread.Start();

       Console.ReadKey();

}}}
```

OUTPUT:

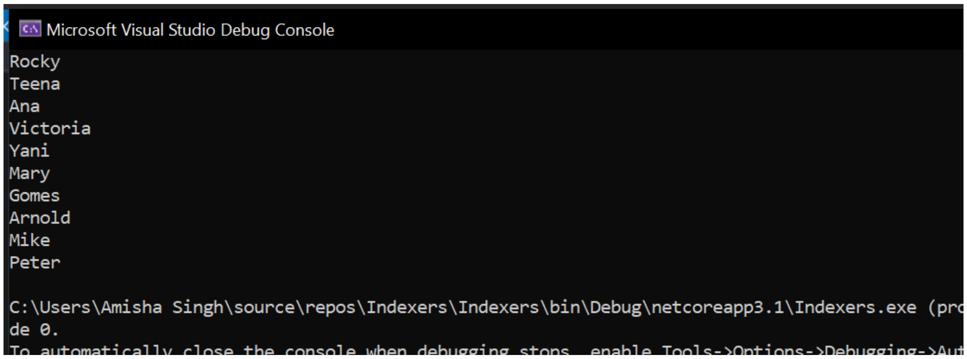


```
In Main: Creating the Child thread
Child thread starts

C:\Users\Amisha Singh\source\repos\Encapsulatio
de 0.
To automatically close the console when debuggi
le when debugging stops.
Press any key to close this window . . .
```

28) Write a program for Indexers.

```
using System;

namespace Indexer_example1

{

    class Program

    {

        class IndexerClass

        {

            private string[] names = new string[10];

            public string this[int i]

            {

                get

                {

                    return names[i];

                }

                set

                {

                    names[i] = value;

                }

            }

        }

        static void Main(string[] args)

        {

            IndexerClass Team = new IndexerClass();

            Team[0] = "Rocky";

            Team[1] = "Teena";

            Team[2] = "Ana";

            Team[3] = "Victoria";

            Team[4] = "Yani";
            Team[5] = "Mary";

            Team[6] = "Gomes";

            Team[7] = "Arnold";

            Team[8] = "Mike";

            Team[9] = "Peter";

            for (int i = 0; i < 10; i++)

            {

                Console.WriteLine(Team[i]);

            }

            Console.ReadKey();

        }

    }

}
```

OUTPUT:

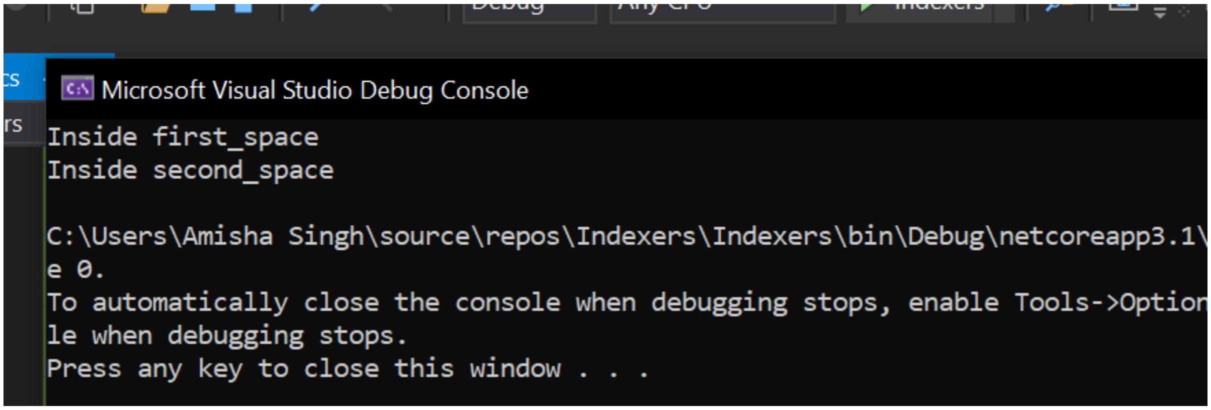


```
Rocky
Teena
Ana
Victoria
Yani
Mary
Gomes
Arnold
Mike
Peter

C:\Users\Amisha Singh\source\repos\Indexers\Indexers\bin\Debug\netcoreapp3.1\Indexers.exe (pro
de 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Aut
```

29) Write a program for Namespace.

```
using System;

namespace first_space

{

class namespace_cl

{

    public void func()

{

       Console.WriteLine("Inside first_space");

   }   }}

namespace second_space

{

class namespace_cl

{

    public void func()

{

         Console.WriteLine("Inside second_space");

      }   }}

class TestClass

{

   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();

   }

}
```

OUTPUT:

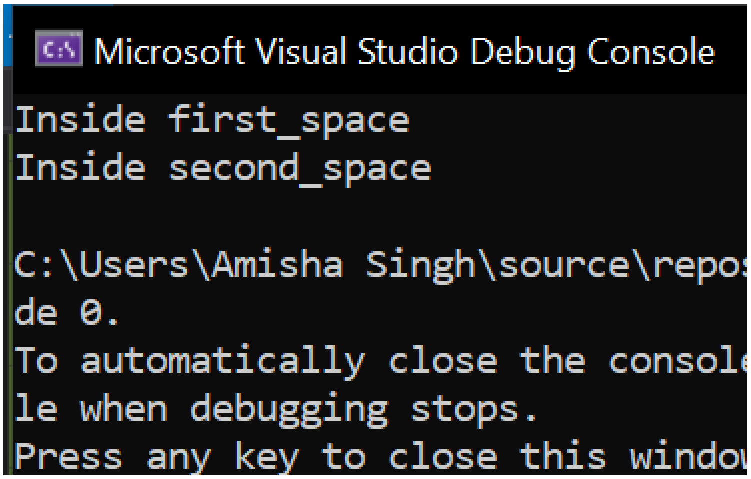


```
Inside first_space
Inside second_space

C:\Users\Amisha Singh\source\repos\Indexers\Indexers\bin\Debug\netcoreapp3.1\
e 0.
To automatically close the console when debugging stops, enable Tools->Option
le when debugging stops.
Press any key to close this window . . .
```

30) Write a program for Constructor.

```
namespace ConstructorDemo

{

class Program

{

Program()

{

Console.WriteLine("This is Constructor");

}

public static void Main(string[] args)

{

Program p = new Program();

Console.WriteLine("Main method");

Console.ReadKey();

}

}

}
```

OUTPUT:

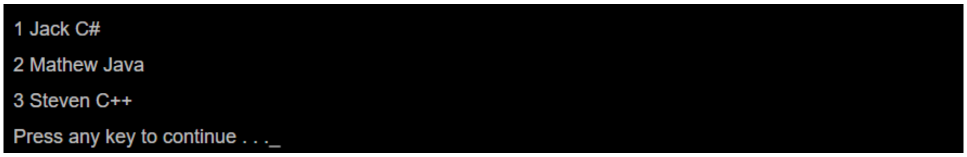


```
Inside first_space
Inside second_space

C:\Users\Amisha Singh\source\repo
de 0.
To automatically close the consol
le when debugging stops.
Press any key to close this windo
```

31) Write a program to access data from database using ADO.NET

```
using System;
using System.Data.SqlClient;

namespace FirstProgram
{
class Program
{
static void Main(string[] args)
{
string ConString = @"Data Source=.\SQLEXPRESS;Initial Catalog=TestDB;Integrated security=True";
SqlConnection con = new SqlConnection(ConString);
string querystring = "Select * from Student";
con.Open();
SqlCommand cmd = new SqlCommand(querystring, con);
SqlDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
Console.WriteLine(reader[0].ToString() + " " + reader[1].ToString() + " " + reader[2].ToString());
}
}
}
}
```

OUTPUT:



```
1 Jack C#
2 Mathew Java
3 Steven C++
Press any key to continue . . ._
```