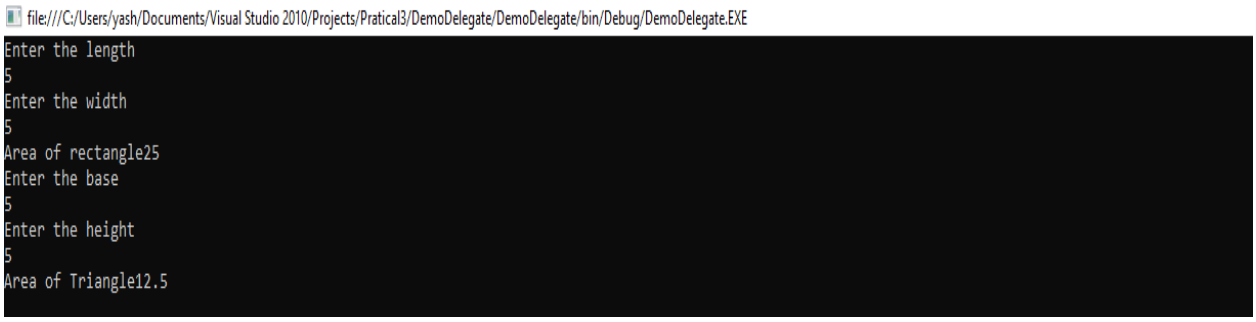
1 Create a delegate calArea(float a,float b) with two float type parameters and having void return type. Create delegate instances for Calculate area of rectangle and triangle and display result on the screen.

```
using System;
namespace DemoDelegate
{
    class Program
    {

     public delegate void calArea(float a, float b);
         public void areaRectangle(float a, float b)
     {
              float ar = a * b;
             Console.WriteLine("Area of
             rectangle"+ar);
     }
     public void areaTriangle(float a, float b)
     {
              float ar = (a * b)/2;
        Console.WriteLine("Area of Triangle" + ar);
    }

    static void Main(string[] args)
    {
        Program de1 = new Program();
        Console.WriteLine("Enter the length");
      float l =
    (float)Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Enter the width");
     float w =
    (float)Convert.ToDouble(Console.ReadLine());
      calArea area = new calArea(de1.areaRectangle);
        area(l, w);
        Console.WriteLine("Enter the base");
     float b =
    (float)Convert.ToDouble(Console.ReadLine());
```

```
            Console.WriteLine("Enter the height");
          float h =
        (float)Convert.ToDouble(Console.ReadLine());
         calArea area1 = new calArea(de1.areaTriangle);
            area1(b,h);
             Console.ReadKey();




        }
    }
}
```

Output:-

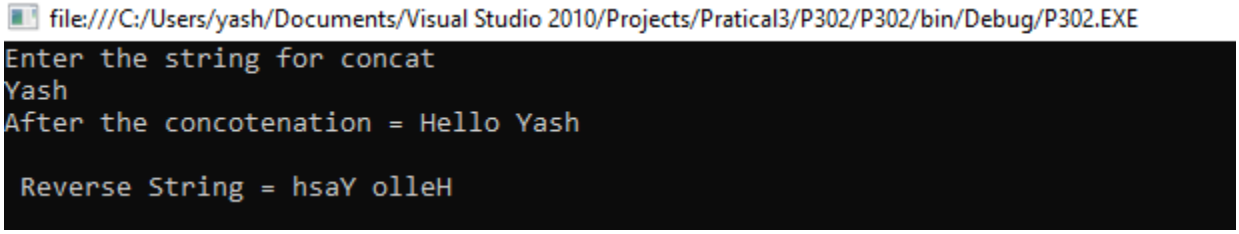


2 Create a delegate with one string parameter and having string return type. Use delegate firstly for concateStr() and secondly use it for reverseStr() method. Create instances of delegate and display concat as well as reverse string by combining delegate instances.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace P302
{
    class StringDemo
    {
```

```
        public delegate string delString(string str);
        public string concateStr(string str)
        {
            return "Hello " + str;
        }
        public string reverseStr(string str)
        {
            char[] rev = str.ToCharArray();
            Array.Reverse(rev);
            return new String(rev);
        }
        static void Main(string[] args)
        {
            StringDemo del = new StringDemo();
             delString strdel1 = new
             delString(del.concateStr);
            Console.WriteLine("Enter the string for
            concat");
            string str = Console.ReadLine();
            string str1 = strdel1(str);
             Console.WriteLine("After the
             concotenation = " + str1);
            delString str2 = new
            delString(del.reverseStr);
            Console.WriteLine("\n Reverse String = " +
            str2(str1));
            Console.ReadKey();
        }
    }
}
```

Output:-



3 Create a program which implements delegate with event model for string modification. Whenever string is modified (by Replace()) fire an event to display a message that is “String is modified”.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EventDelegate
{
    class EventDelegate
    {
        public delegate void delModified();
        public static event delModified modify;
        public static void strChange()
        {
            Console.WriteLine("Enter the string");
            string str = Console.ReadLine();
             Console.WriteLine("String
             replace"+str.Replace("a","*"));
            Console.WriteLine("***String is modified
            ***");

        }
        static void Main(string[] args)
        {
             EventDelegate eventdel = new
             EventDelegate();
            modify = new delModified(strChange);
            modify.Invoke();
            Console.ReadKey();
        }
    }
```

```
}
```

Output:-

4 Create a program which Demonstrate file read and write operation using file stream.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
namespace File
{
    class Program
    {
        static void Main(string[] args)
        {
             string[] names = new string[] { "Yash",
             "xyz", "pqr" };
            using (StreamWriter sw = new
            StreamWriter("file1.txt"))
            {
                foreach (string s in names)
                {
                    sw.WriteLine(s);
                }
            }
            string line = "";
             using (StreamReader sr = new
             StreamReader("file1.txt"))
            {
                while ((line = sr.ReadLine()) != null)
                {
                    Console.WriteLine(line);
```

```
                }
            }
            Console.ReadKey();
        }
    }
}
```

Output:-

5 Create a program which Demonstrate use of regular expression for emil and phone number.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
namespace RegulareExp
{
    class Program
    {
        static void Main(string[] args)
        {
             string eailp = @"^\w+@[a-zA-Z_]+?\.[a-zA-
             z]{2,3}$";
            string phonp = @"^\d{10}$";
            Console.WriteLine(" Enter Email address");
            string emil = Console.ReadLine();
             bool EmailValid = Regex.IsMatch(emil,
             eailp);
            Console.WriteLine(" Enter phone number");
            string phone= Console.ReadLine();
             bool phoneValid = Regex.IsMatch(phone,
             phonp);
```

```
            if (!EmailValid)
            {
                Console.WriteLine("Email is not valid");
            }
            else
            {
                Console.WriteLine("Email id is valid");
            }
            if (!phoneValid)
            {
                Console.WriteLine("Phone is not valid");
            }
            else
            {
                Console.WriteLine("Phone is valid");
            }
            Console.ReadKey();

        }
    }
}
```