Practical – 4 Delegates and events

Q-1. Create a delegate Cal Area(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.

Code:-

```
ConsoleApp1                                    CalculateAreaDelegate
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   delegate void CalculateAreaDelegate(float a, float b);
8   namespace ConsoleApp1
9   {
10      internal class Program
11      {
12          static void CalculateRectangleArea(float length, float width)
13          {
14              float area = length * width;
15              Console.WriteLine($"Area of rectangle: {area}");
16          }
17
18          static void CalculateTriangleArea(float baseLength, float height)
19          {
20              float area = 0.5f * baseLength * height;
21              Console.WriteLine($"Area of triangle: {area}");
22          }
23          static void Main(string[] args)
24          {
25              CalculateAreaDelegate rectangleAreaDelegate = new CalculateAreaDelegate(CalculateRectangleArea);
26              CalculateAreaDelegate triangleAreaDelegate = new CalculateAreaDelegate(CalculateTriangleArea);
27
28              float rectangleLength = 5.0f;
29              float rectangleWidth = 3.0f;
30              rectangleAreaDelegate(rectangleLength, rectangleWidth);
31
32              float triangleBaseLength = 4.0f;
33              float triangleHeight = 6.0f;
            float triangleHeight = 6.0f;
            triangleAreaDelegate(triangleBaseLength, triangleHeight);

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

Output:-

```
Area of rectangle: 15
Area of triangle: 12
```

Practical – 4 Delegates and events

Q-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.

Code:-

```
leApp2                                  ConsoleApp2.Program                    Main(string[] args)
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Text;
 5  using System.Threading.Tasks;
 6
 7  public delegate string StringOperationDelegate(string input);
 8
 9  namespace ConsoleApp2
10  {
11      public class StringManipulator
12      {
13          public string ConcatenateStrings(string input)
14          {
15              return "Concatenated: " + input;
16          }
17
18          public string ReverseString(string input)
19          {
20              char[] charArray = input.ToCharArray();
21              Array.Reverse(charArray);
22              return new string(charArray);
23          }
24      }
25
leApp2                                  ConsoleApp2.Program                    Main(string[] args)
25
26      internal class Program
27      {
28          static void Main(string[] args)
29          {
30              StringManipulator manipulator = new StringManipulator();
31
32              StringOperationDelegate concatenateDelegate = new StringOperationDelegate(manipulator.ConcatenateStrings);
33              StringOperationDelegate reverseDelegate = new StringOperationDelegate(manipulator.ReverseString);
34
35              string inputStr = "Hello, World!";
36              string concatenatedStr = concatenateDelegate(inputStr);
37              string reversedStr = reverseDelegate(inputStr);
38
39
40              Console.WriteLine(concatenatedStr);
41              Console.WriteLine(reversedStr);
42
43              Console.ReadLine();
44
45
46          }
47      }
48  }
49
```

Practical – 4 Delegates and events

Output:-

```
Concatenated: Hello, World!
!dlroW ,olleH
```

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.

Code:-

```
leApp3                    ConsoleApp3.Program                    StringModifiedEventHandler(object
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Text;
 5  using System.Threading.Tasks;
 6  using static ConsoleApp3.StringModifier;
 7
 8  namespace ConsoleApp3
 9  {
10      public class StringModifier
11      {
12          public delegate void StringModifiedEventHandler(object sender, EventArgs e);
13
14          public event StringModifiedEventHandler StringModified;
15
16          private string modifiedString;
17
18          public string ModifiedString
19          {
20              get => modifiedString;
21              private set
22              {
23                  if (modifiedString != value)
24                  {
25                      modifiedString = value;
26
27                      // Raise the event when the string is modified
28                      OnStringModified();
29                  }
30              }
33          public StringModifier(string initialString)
34          {
35              ModifiedString = initialString;
36          }
37
38          public void Replace(string oldValue, string newValue)
39          {
40              ModifiedString = ModifiedString.Replace(oldValue, newValue);
41          }
42
43          // Event raising method
44          protected virtual void OnStringModified()
45          {
46              StringModified?.Invoke(this, EventArgs.Empty);
47          }
48
49      }
```

Practical – 4 Delegates and events

```
internal class Program
{
    static void Main(string[] args)
    {
        StringModifier stringModifier = new StringModifier("Hello, World!");

        stringModifier.StringModified += StringModifiedEventHandler;

        stringModifier.Replace("Hello", "Hi");

        Console.WriteLine("Press any key to exit...");
        Console.ReadKey();
    }
    static void StringModifiedEventHandler(object sender, EventArgs e)
    {
        Console.WriteLine("String is modified");
    }
}
```

Output:-

```
String is modified
Press any key to exit...
```

4. Write a program to create a delegate called TrafficDel and a class called TrafficSignal with the following delegate methods.

```
Public static void Yellow() {

Console.WriteLine(Yellow Light Signal To Get Ready);

}

Public static void Green() {

Console.WriteLine(Green Light Signal To Go);

}

Public static void Red() {

Console.WriteLine(Red Light Signal To Stop);

}
```

Practical – 4 Delegates and events

Also include a method IdentifySignal() to initialize an array of delegate with the above methods and a method show() to invoke members of the above array.

Code:-

```
oleApp4                                  ConsoleApp4.Program                    IdentifySignal()
 1   using System;
 2   using System.Collections.Generic;
 3   using System.Linq;
 4   using System.Text;
 5   using System.Threading.Tasks;
 6
 7   public delegate void TrafficDel();
 8
 9   namespace ConsoleApp4
10   {
11       public class TrafficSignal
12       {
13           public static void Yellow()
14           {
15               Console.WriteLine("Yellow Light Signal To Get Ready");
16           }
17
18           public static void Green()
19           {
20               Console.WriteLine("Green Light Signal To Go");
21           }
22
23           public static void Red()
24           {
25               Console.WriteLine("Red Light Signal To Stop");
26           }
27       }
```

```
oleApp4                                  ConsoleApp4.Program                    IdentifyS
28       internal class Program
29       {
30           public static TrafficDel[] IdentifySignal()
31           {
32               TrafficDel[] signals = new TrafficDel[3];
33               signals[0] = TrafficSignal.Yellow;
34               signals[1] = TrafficSignal.Green;
35               signals[2] = TrafficSignal.Red;
36               return signals;
37           }
38
39           public static void Show(TrafficDel[] signals)
40           {
41               foreach (TrafficDel signal in signals)
42               {
43                   signal();
44               }
45           }
46
47           static void Main(string[] args)
48           {
49               TrafficDel[] signals = IdentifySignal();
50               Show(signals);
51               Console.ReadLine();
52           }
53       }
54   }
55
```

Practical – 4 Delegates and events

Output:-

```
Yellow Light Signal To Get Ready
Green Light Signal To Go
Red Light Signal To Stop
```