Assignment On

Web Computing and Mining (MCSE-541)

Assignment – 04

Submitted to

Dr. Md Shamim Akhter

Submitted by:

Mohammed Morad Hossen

Student ID: MCE 079 055 32

Dept of Computer Science and Engineering

Stamford University Bangladesh

Assignment 4 : Task 1 Solution

```
using System;
using System.Collections.Generic;

namespace Assignment4
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            List<Employee> employeeList = new List<Employee>();

            employeeList.Add(new Employee() { name = "Merinda", id = 1, salary = 200000.00, experience = 5 });
            employeeList.Add(new Employee() { name = "Belal", id = 2, salary = 255000, experience = 4 });
            employeeList.Add(new Employee() { name = "Roy", id = 3, salary = 280000, experience = 6 });
            employeeList.Add(new Employee() { name = "Poly", id = 4, salary = 160000, experience = 3 });

            Employee.isPromotable(employeeList);

        }
    }
```

```
    public class Employee
    {
        5 references
        public string name { get; set; }
        4 references
        public int id { get; set; }
        4 references
        public double salary { get; set; }
        5 references
        public int experience { get; set; }

        1 reference
        public static void isPromotable(List<Employee> emplist)
        {
            foreach (Employee e in emplist)
                if (e.experience > 4)
                    Console.WriteLine("The promoted employee is  " + e.name);


        }
    }
}
```

Output

```
Microsoft Visual Studio Debug Console
The promoted employee is  Merinda
The promoted employee is  Roy

C:\Users\MORAD\source\repos\Assignment4\Assignment4\bin\Debug\netcoreapp3.1\Assignment4.exe (process 10032) 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 . . .
```

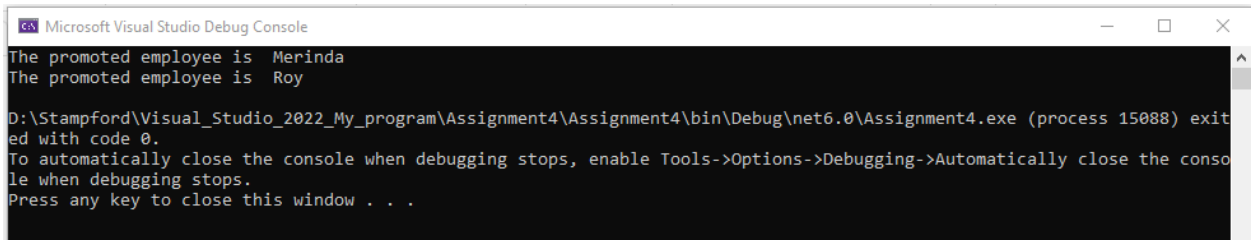
Task 1 (a) and (b) Solution:

```
using System;
using System.Collections.Generic;

namespace Assignment4
{
    public delegate bool PDelegate (Employee e);
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            List<Employee> employeeList = new List<Employee>();

            employeeList.Add(new Employee() { name = "Merinda", id = 1, salary = 200000.00, experience = 5 });
            employeeList.Add(new Employee() { name = "Belal", id = 2, salary = 255000, experience = 4 });
            employeeList.Add(new Employee() { name = "Roy", id = 3, salary = 280000, experience = 6 });
            employeeList.Add(new Employee() { name = "Poly", id = 4, salary = 160000, experience = 3 });

            // Employee.isPromotable(employeeList);
```

```
            PDelegate dm = isPromotable;
            Employee.Promotable(employeeList, dm);


        }

        1 reference
        public static bool isPromotable(Employee e)
        {
            if (e.experience > 4)
                return true;
            else return false;

        }

    }

    11 references
    public class Employee
    {
        5 references
        public string name { get; set; }
        4 references
        public int id { get; set; }
        4 references
        public double salary { get; set; }
        5 references
        public int experience { get; set; }
```

```
41
                1 reference
42              public static void Promotable(List<Employee> emplist, PDelegate pd )
43              {
44                  foreach (Employee e in emplist)
45                  {
46                      if (pd(e))
47                          Console.WriteLine("The promoted employee is  " + e.name);
48                  }
49              }
50          }
51      }
52
```

Output:



Task 3 Solution:

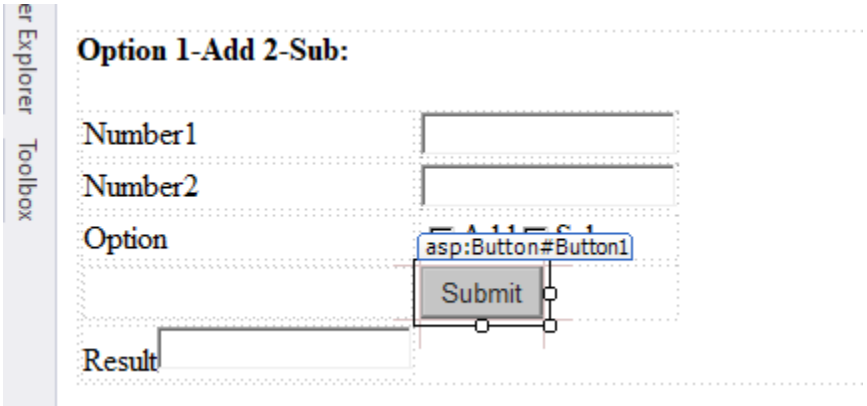
Webform1

Client Objects & Events | (No Events)

```
1   <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Assignment4Web_App.WebForm1" %>
2
3   <!DOCTYPE html>
4
5   <html xmlns="http://www.w3.org/1999/xhtml">
6   <head runat="server">
7       <title></title>
8   </head>
9   <body>
10      <form id="form1" runat="server">
11          <div>
12              <h4>  Option 1-Add 2-Sub:  </h4>
13              <table class="auto-style3">
14
15                  <tr>
16                      <td class="auto-style3">
17                          <asp:Label ID ="Lable12" runat="server" Text="Number1"></asp:Label> </td>
18                      <td>
19                          <asp:TextBox  ID="TextBox1" runat="server" CssClass="auto-style2"> </asp:TextBox>
20                      </td>
21
22                  </tr>
23
24              <tr>
25                      <td class="auto-style3">
26                          <asp:Label ID ="Label2" runat="server" Text="Number2"></asp:Label> </td>
27                      <td>
28                          <asp:TextBox  ID="TextBox2" runat="server" CssClass="auto-style2"> </asp:TextBox>
29                      </td>
```

```
                </tr>

            <tr>
                <td class ="auto-style3">
                    <asp:Label ID="Lable3" runat= "server" Text="Option"></asp:Label>
                    </td>
                <td>
                    <asp:CheckBox ID="CheckBox1" runat="server" Text="Add" />
                    <asp:CheckBox ID ="CheckBox2" runat="server" Text="Sub" />
                </td>
            </tr>
                 <tr>
                <td> </td>

                <td class="auto-style3">
                    <asp:Button ID="Button1" runat="server" Text="Submit" OnClick="Button_Classs" />

                </td>

                </tr>
                <tr>
                    <td class="auto-style3">
                        <asp:Label ID="Lable4" runat="server" Text="Result" ></asp:Label>
                        <asp:TextBox ID="TextBox3" runat="server" CssClass="auto-style2"></asp:TextBox>

                     </td>
                    </tr>

                  </table>
        </div>
    </form>
</body>
</html>

```

Submit Button Code

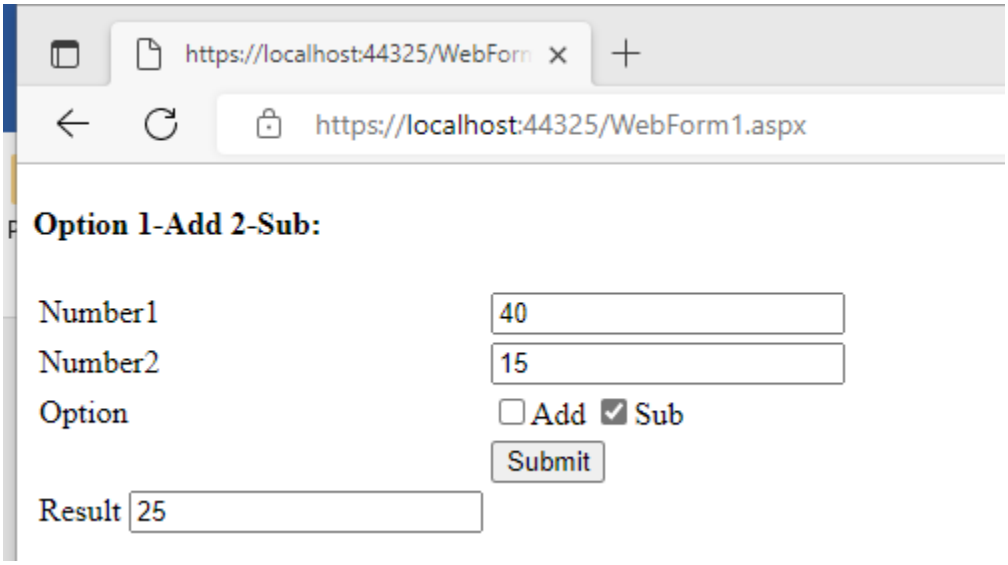


```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Web;
5   using System.Web.UI;
6   using System.Web.UI.WebControls;
7
8   namespace Assignment4Web_App
9   {
        2 references
10      public partial class WebForm1 : System.Web.UI.Page
11      {
            0 references
12          protected void Page_Load(object sender, EventArgs e)
13          {
14
15          }
16
            0 references
17          protected void Button_Classs(object sender, EventArgs e)
18          {
19              int result = 0; int opt = 1;
20              int number1 = Convert.ToInt32(TextBox1.Text);
21              int number2 = Convert.ToInt32(TextBox2.Text);
22
23              if (CheckBox1.Checked)
24              {
25                  opt = 1;
26                  CheckBox2.Checked = false;
27              }
28
29              if (CheckBox2.Checked)
30              {
31                  opt = 2;
32                  CheckBox1.Checked = false;
33
34              }
35
36              if (opt == 1) result = MathClass.Add(number1, number2);
37              if (opt == 2) result = MathClass.Sub(number1, number2);
38
39              TextBox3.Text = result.ToString();
40
41          }
42      }
43  }
```

MathClass.cs

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Web;
5
6   namespace Assignment4Web_App
7   {
        2 references
8       public class MathClass
9       {
            1 reference
10          public static int Add (int x, int y)
11
12          {
13              return x + y;
14          }
15
            1 reference
16          public static int Sub( int x, int y)
17          {
18              return x - y;
19          }
20      }
21  }
```

Output:



Task 3 (loosing the classes a little bit)

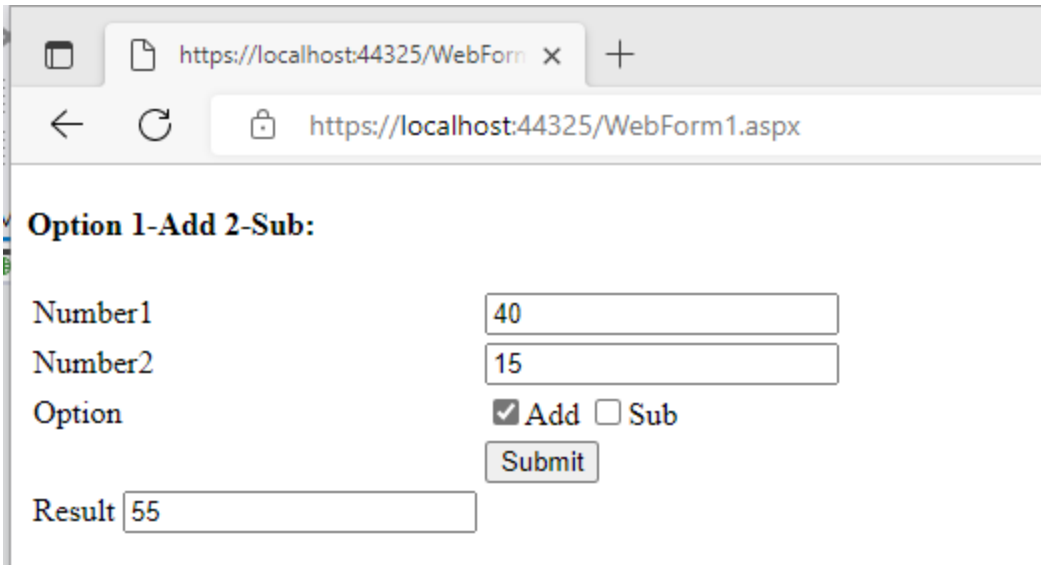
Submit Button Code

```
36              //if (opt == 1) result = MathClass.Add(number1, number2);
37              // if (opt == 2) result = MathClass.Sub(number1, number2);
38
39              result = MathClass.getPointer(opt, number1, number2);
40
41              TextBox3.Text = result.ToString();
42
43          }
44      }
45  }
```

Math Class code

```
        1 reference
21      public static int getPointer (int operation, int number1, int number2)
22      {
23          if (operation == 1)
24
25              return Add(number1, number2);
26
27          else if (operation == 2)
28
29              return Sub(number1, number2);
30
31          else return 0;
32
33      }
34  }
35  }
```

Output::



Loosing Classes a more little bit

Submit Button code

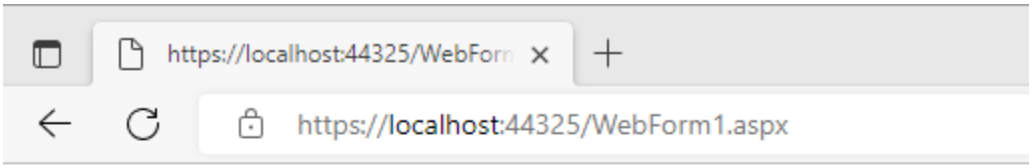
```
36                  //if (opt == 1) result = MathClass.Add(number1, number2);
37                  // if (opt == 2) result = MathClass.Sub(number1, number2);
38
39                  //  result = MathClass.getPointer(opt, number1, number2);
40
41                  result = MathClass.getPointer(opt).Invoke (number1, number2);
42
43                  TextBox3.Text = result.ToString();
44
45              }
46          }
47      }
```

Math Class code

```
6   namespace Assignment4Web_App
7   {
8       public  delegate  int pointerFunction(int x, int y);
            1 reference
```

```
1 reference
public static pointerFunction getPointer (int operation)
{
    pointerFunction pf = null;
    if (operation == 1)
         pf=Add;

    else if (operation == 2)
         pf=Sub;

     return pf;

}
}
}
```

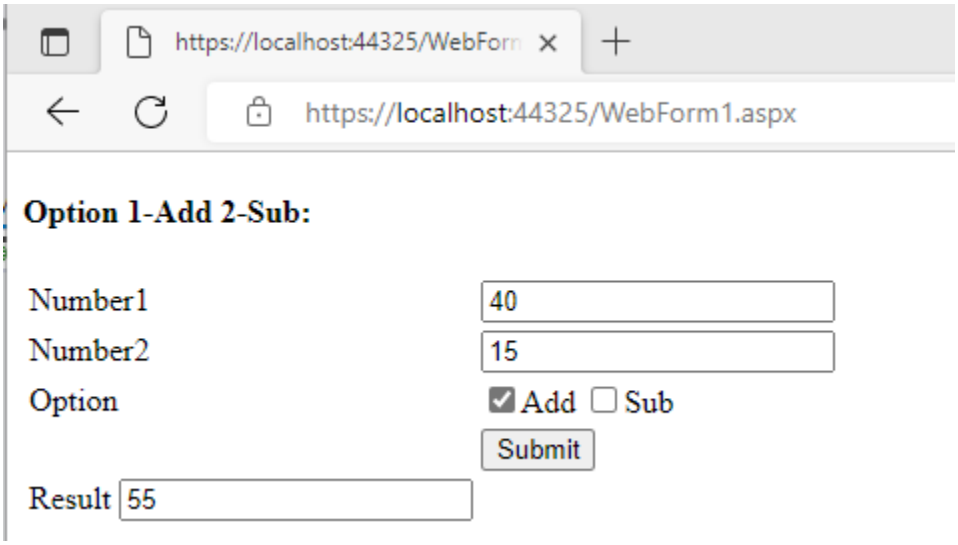
Output:



Anonymous Function (MatchClass)

Output:

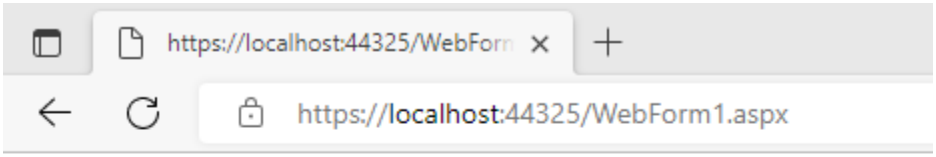
```
22          1 reference
            public static pointerFunction getPointer (int operation)
23          {
24              pointerFunction pf = null;
25              if (operation == 1)
26                   pf=delegate (int x, int y)
27                     {
28                  return x + y;
29                     };
30
31              else if (operation == 2)
32                   pf= delegate (int x, int y)
33                   {
34                       return x - y;
35                   }; ;
36
37                return pf;
38
39          }
40      }
41  }
```



Lambda Expression in MathClass

```
1 reference
24          public static pointerFunction getPointer (int operation)
25          {
26              pointerFunction pf = null;
27              if (operation == 1)
28                   pf= (int x, int y) =>
29                     {
30                  return x + y;
31                     };
32
33              else if (operation == 2)
34                   pf=  (int x, int y) =>
35                   {
36                       return x - y;
37                   };
38
39               return pf;
40
41          }
42      }
43  }
```

Output:



Option 1-Add 2-Sub:

Number1 40

Number2 15

Option ☑ Add ☐ Sub

Submit

Result 55

Lambda expression in shorter form

```
1 reference
24          public static pointerFunction getPointer (int operation)
25          {
26              pointerFunction pf = null;
27              if (operation == 1)
28                   pf= (int x, int y) => x + y;
29
30
31              else if (operation == 2)
32                   pf=  (int x, int y) => x - y;
33
34               return pf;
35
36          }
37      }
38  }
```

Output: