

Web Application Development-I

(Using C#)

Subject Code : CC-301 with Practical CC-304

Prof. Kamesh Raval

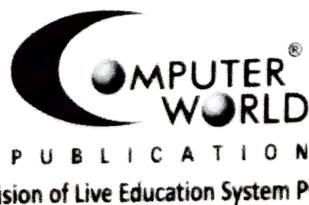
MCA [Master in Computer Application]
Som-Lalit Institute of Computer Application,
Ahmedabad.

Prof. Nirav S. Shukla

MCA [Master in Computer Application]
Chimanbhai Patel Institute of Computer Applications,
SG Highway, Ahmedabad.

Dr. Sanjay B. Sonar

PhD [Computer Science]
Navgujarat College of Computer Applications,
Ashram Road, Ahmedabad.



COMPUTER WORLD

43, 5th Floor, SANIDHYA Complex, Nr. M. J. Library,
Opp. Sanyas Ashram, Ashram Road, Ahmedabad-09.

Ph. : +91- 79 26580723, 26580823 | Mobile : 9724011150, 9725022917
URL : www.computerworld.ind.in | e-Mail : info@computerworld.ind.in

Publish By :

COMPUTER World

A Division of Live Education System Pvt. Ltd.

An ISO 9001 : 2008 Certified Company

Kalpesh Patel - kalpesh@computerworld.ind.in

Prepare by : Computer World Research Department

Prof. Kamesh Raval, Prof. Nirav S. Shukla,

Dr. Sanjay B. Sonar, Ravindra Parmar

Price : ₹ 199/-

Book Code : CEBCA119

Edition : 1st

Notice of Rights

No part of this publication may be reproduced, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of ComputerWorld except under the terms of a ComputerWorld license agreement.

Notice of Liability

The information in this courseware title is distributed on an 'as is' basis, without warranty. While very precaution has been taken in the preparation of this course, neither the authors nor ComputerWorld shall have any liability to my person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instruction contained in this book or by the computer software and hardware products described in it.

Disclaimer

We make a sincere effort to ensure the accuracy of the material described here in however, ComputerWorld makes no warranty, expressed or implied, with respect of the quality, correctness, reliability, accuracy, of freedom from error of this document or the products it describes. Data used in examples and sample data files are intended to be fictional. Any resemblance to real persons or companies is entirely coincidental.

Content of Copyright : COMPUTER WORLD PUBLICATION - <http://www.computerworld.ind.in>

Contact : 43, 5th Floor, Sanidhya Complex, Opp. Capital Comm. Center, Ashram Road, Ahmedabad-09

Phone : +91-79-2658 0823 / +91-79-2658 0723, | Mobile : 97240 11150, 97250 22917, 97250 19114

URL : www.computerworld.ind.in | e-Mail : info@computerworld.ind.in, info.computerworld@gmail.com

INDEX**Unit – 1 Introduction to ASP.NET****05 to 64**

- 1.1 Introduction to ASP.NET
 - 1.2 Working with ASP.NET
 - 1.3 Application Configuration
 - 1.4 Common Properties
 - 1.5 Label Control
 - 1.6 Textbox Control
 - 1.7 Literal Control
 - 1.8 Panel control
- ❖ Unit-1 CC-304 Practical

Unit – 2 Information Passing and Standard Controls**65 to 110**

- 2.1 PASSING INFORMATION FROM ONE PAGE TO ANOTHER
 - 2.2 Web Server Controls Button
 - 2.3 Hyperlink control
 - 2.4 Image map control
 - 2.5 Checkbox control
 - 2.6 RadioButton Control
 - 2.7 CheckBoxList Control
 - 2.8 RadioButtonList Control
 - 2.9 ListBox Control
 - 2.10 DropDownList Control
 - 2.11 FileUpload Control
 - 2.12 AdRotator Control
 - 2.13 Calendar Control
- ❖ Unit-2 CC-304 Practical

Unit – 3 Master page, Navigation, Validation, Login controls

111 to 173

3.1 Designing Master page**3.2 Using Navigation Controls**

3.2.1 TreeView Control

3.2.2 SiteMapPath

3.2.3 Menu control

3.2.4 Creating sitemap file for navigation

3.3 Validation controls

3.3.1 Required Field Validator

3.3.2 Range validator

3.3.3 Regular Expression validator

3.4 Login Controls

3.4.1 Login

3.4.2 LoginView and LoginName control

3.4.3 PasswordRecovery

3.4.4 LoginStatus

3.4.5 CreateUserWizard

3.4.6 ChangePassword

3.5 Creating and Managing Roles**3.6 Creating and Managing Access Rules****3.7 Creating and Managing Profile**

❖ Unit-3 CC-304 Practical

Unit – 4 Authentication, Authorization and Data Controls

174 to 240

4.1 Membership class

4.2 Role class

4.3 ProfileManager class

4.4 DataList Control

4.5 ADO.NET

❖ Unit-4 CC-304 Practical

❖ Paper 2019

241 to 244

Unit -1 Introduction to ASP.NET

1.1: Introduction to ASP.NET:

➤ INTRODUCTION:

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.

Active Server Pages or ASP, as it is more commonly known, is a technology that enables you to make dynamic and interactive web pages. ASP uses server-side scripting to dynamically produce web pages that are not affected by the type of browser the web site visitor is using. The default scripting language used for writing ASP is VBScript, although you can use other scripting languages like JScript (Microsoft's version of JavaScript).

➤ EXECUTION MODEL OF ASP.NET:

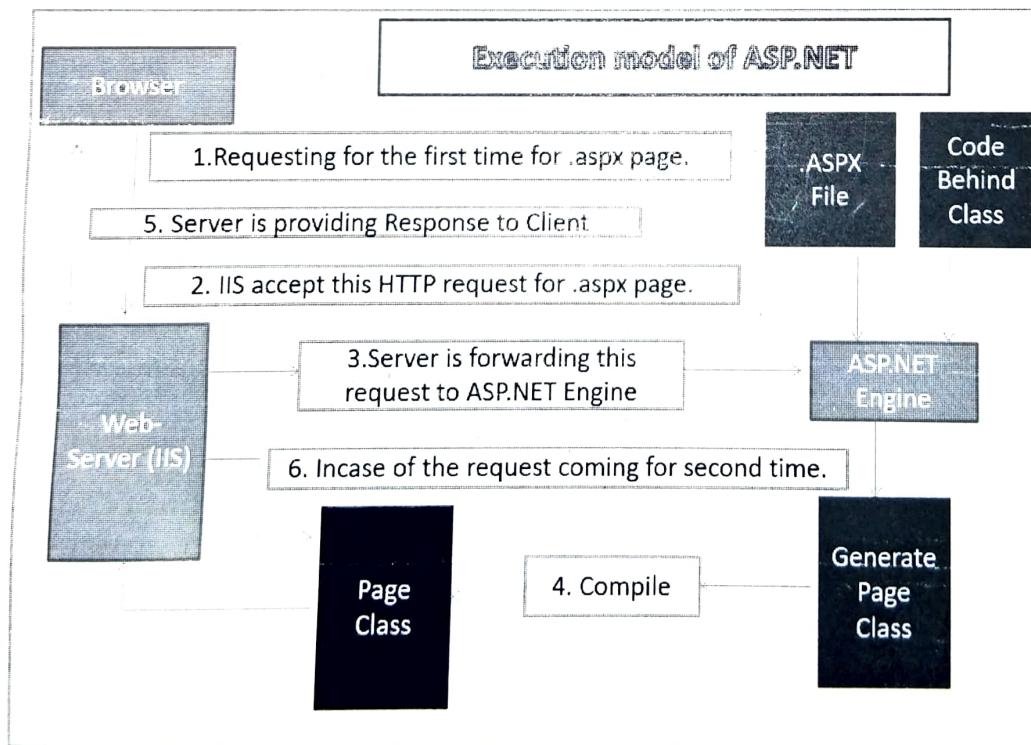


Figure 1.1: ASP.NET Execution Model

1. In the first step, User can send a request to particular .aspx page to the web server. Here browser will be treated as a client and a machine on which IIS (Internet Information Server) is installed will act as a web-server. The communication between client and server is done using HTTP (Hypertext Transfer Protocol).
2. In the second step web-server will accept the request coming from client (web-browser). Web server will search the requested page and call ASP.NET engine.
3. Web-page have designer code (in XHTML language) which is stored in '.aspx' page and code (in C#.NET language) which is stored in '.aspx.cs' page. ASP.NET engine now merge both the partial class file to generate complete class file.
4. This complete (merged) class file will be compiled by ASP.NET and compiled version of class file will be stored in the cache.
5. Web-server now retrieve the class file and send the response to the web-browser, who has requested the page.
6. If the request is coming again for the same web page, then web-server will retrieve the compiled version of web page from the cache and give the response to the web-browser.

➤ **ASP.NET Framework:**

ASP.Net Framework primarily consists of two main components:

1. **Common Language Runtime (CLR):** This is a language-independent runtime environment of .NET Framework that manages code at execution time. Irrespective of what language chooses to develop your application, CLR manages the code for all.
2. **Framework Class Library (FCL):** This is a repository of built-in classes and interfaces that corresponds to a specific functionality. FCL provides thousands of readily available classes to the website developer, so that developer can make the objects of the classes available in the library and use the functionalities of different classes. This saves lots of time and effort of the developer.

➤ **IIS (Internet Information Server):**

IIS stands for Internet information Server. It is a web-server, use to process ASP.NET webpages. IIS web-server comes free as bundle with Microsoft's Windows operating systems.

IIS is very rich with features. Most common features provided by IIS are:

IIS is used to host static websites as well ASP.NET dynamic websites. It can also support for an FTP server, host WCF services, and be extended to host web applications built on other platforms such as PHP.

There are many built-in authentication options such as Basic, ASP.NET, and Windows auth. The latter is useful if you have a Windows Active Directory environment—users can be automatically signed into web applications using their domain account. Other

built-in security features include TLS certificate management and binding for enabling HTTPS and SFTP on your sites, request filtering for whitelisting or blacklisting traffic, authorization rules, request logging, and a rich set of FTP-specific security options.

Application pool is one of the key features of IIS. It also supports remote management. IIS can also be managed using the CLI or using PowerShell. You can script everything, which is great if you like the power that comes with being able to do so.

➤ Features of ASP.NET:

Compare to classical ASP (Active Server Pages), ASP.NET has include the following features:

- **64-Bit Support:** ASP.NET supports 64-bit supports. If the ASP.NET website is made in any 32-bit machine or 64-bit machine, it can be hosted on any IIS based web-server without any kind of version conflicts. IIS can also be installed in any system regardless it is of 32-bit Windows OS or 64-bit Windows OS.
- **ACL Support:** ASP.NET provide a support for ACL (Access Control List). Using ACL web developers can easily implement security with very less effort.
- **New features in ADO.NET:** In ASP.NET, abstract layer control is designed. Web developers, just need to drag and drop the control and configure it with very few wizard steps. One control includes connection, data adapter, dataset and command builder as a package. This will give freedom to web developer from writing huge amount of code managing connectivity with the databases.
- **FTP Support:** ASP.NET comes up with built-in FTP (File Transfer Protocol) support. With this feature, web developers can open the websites into their machines directly from the web-server and can do any type of modifications into the web pages on live site.
- **The New Ping class:** The feature of Ping class provided by the ASP.NET is used to test network connectivity.
- **SMTP Support:** ASP.NET supports SMTP (Simple Mail Transfer Protocol). SMTP is used to send mail. Developers of the website just need to declare the object of SmtpClient class, set the host and port property and call send method to send any mail.
- **Generics:** ASP.NET also supports Generic classes.

1.2 Working with ASP.NET:

➤ IDE of ASP.NET:

IDE stands for Integrated Development Environment. It is a software which provides GUI based, easy development environment to the web-developer of ASP.NET websites. The IDE used to develop ASP.NET websites is called VWD (Visual Web Developer). The features of VWD are listed as below:

- It provides drag and drop interface.

- It has code editor with rich intelliSense functionality.
- It also has integrated access to SQL Server.
- It provides multi language support.
- Built in web-server is also embedded in the Visual Web Developer.
- It supports dynamic compilation.
- Visual Web Developer has very good error notification assistance.
- It also includes facility to validate HTML codes written with different HTML versions.
- It provides built-in code snippets.

➤ **Coding models of ASP.NET:**

ASP.NET coding models are the methodologies used for writing code in a web application.

ASP.NET provides two coding models:

1. Inline Coding Model
2. Code Behind Model

[1] Inline Coding Model:

- Inline Coding model was introduced prior to the code-behind model.
- In this model, the code required to implement a web page is directly placed in the .aspx file on application.
- The major drawback of this coding model is that, you need to write the code to implement various functionalities of a web application and the HTML code in a common file.

[2] Code-Behind Coding Model:

- To deal with this drawback of Inline Code model, ASP.NET 1.0 introduced the code behind model.
- In the Code Behind Model you have to maintain separate code file for each web page. It uses concept of Partial classes.
- One file stores the code to implement the functionalities of a web page, and other file stores the HTML coding of web application.
- Suppose that we have created a web page called “default” then in the Code-Behind Model HTML code will be stored in default.aspx file and VB Code will be stored in default.aspx.vb file.
- If the code written in C# language instead of VB lang. then this code will be stored in “default.aspx.cs” file.

Type of ASP.NET Pages	Files created in Inline coding model	Files created in Code-Behind coding model
Web form	.aspx	.aspx .aspx.cs / .aspx.vb
Master page	.master	.master .master.cs / .master.vb
Web user control	.ascx	.ascx .ascx.cs / .ascx.vb
Web services	.asmx	.asmx .asmx.cs / asmx.vb

➤ **Code-Render Block:**

Code render blocks are used to write inline code or inline expression inside the .aspx page. This inline code is executed by server and displays its output on the client by the browser. The code render blocks are represented on a page by <% %> symbols. Code present inside these symbols are executed in top-down manner.

For Example, consider the following code segment, in which we have used loop control to increase the size of font in the <Body> tag:

```
<%@ Page Language="C#" AutoEventWireup="true"
   CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
   Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
   transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
   <title>Welcome Page</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <% int i;
            for (i=0; i<8; i++)
            { %>
```

```
<font size="<%="i %>">WelCome to ASP.Net</font><br />
<% }%>
</div>
</form>
</body>
</html>
```

➤ **Directory Structure of ASP.NET:**

In ASP.NET, there are some specific directories are there, which have some specific use. Microsoft has suggested to include a specific type of file you need to use specific built-in directory. The following directories are there in the directory structure of ASP.NET.

1. App_Data
2. App_Code
3. Bin
4. App_Themes
5. App_GlobalResources
6. App_Browser

[1] \App_Data:

- This folder holds the data stores utilized by the application.
- It is good to spot to centrally store all the data stores your application might use.
- This directory can contain .mdf (Microsoft SQL Server files) or .mdb (Microsoft Access Files).

[2] \App_Code:

- By default, ASP.NET 2.0 provides a unique directory known as App_Code to store the code files that are accessible all the web pages of an application.
- Once you add something to App_Code directory then it is automatically detected and complied. Once complied then any page in application can access them.

[3] \Bin:

- The code for a specific functionality in a web can also made accessible to all web pages of a Web application by storing the sharable files in the bin directory.
- ASP.NET 1.x supports this method for code sharing.
- The bin directory is similar to App_code directory except it stores the precompiled assemblies of the code files.

➤ Life-cycle of Web Pages:

When the request is made by the browser for specific webpage, then series of events will be fired and number of subroutines will be executed by the server. After receiving the request, the page gets constructed in the memory; it is called 'Initialization' of the page. Once the page is initialized then 'Load' event will be fired, where the various .NET controls is placed on the page. After Load event, the gets 'Render', In this event the .NET server controls are translated in the equivalent HTML controls. The following series of events will be executed by the server is called the life-cycle of the ASP.NET web page.

- **PreInit** - PreInit is the first event in page life cycle. It checks the IsPostBack property and determines whether the page is a postback or not. It also sets the themes and applies the master pages, creates dynamic controls will be also placed during this event, and gets and sets profile property values. This event can be handled by overloading the OnPreInit method or creating a Page_PreInit handler.
- **Init** - Init event initializes the control property and built the control tree. This event can be handled by overloading the OnInit method or creating a Page_Init handler.
- **InitComplete** - InitComplete event manages view states. All the controls turn on view-state tracking.
- **LoadViewState** - LoadViewState event allows loading view state information into the controls.
- **LoadpostData** - During this phase, the contents of all the input fields are defined with the <form> tag are processed.
- **PreLoad** - PreLoad occurs before the post back data is loaded in the controls. This event can be handled by overloading the OnPreLoad method or creating a Page_PreLoad handler.
- **Load** - The Load event is raised for the page first and then recursively for all child controls. The controls in the control tree are created. This event can be handled by overloading the OnLoad method or creating a Page_Load handler.
- **LoadComplete** - The loading process is completed, control event handlers are run, and page validation takes place. This event can be handled by overloading the OnLoadComplete method or creating a Page_LoadComplete handler.
- **PreRender** - The PreRender event occurs just before the output is rendered. By handling this event, pages and controls can perform any updates before the output is rendered.
- **PreRenderComplete** - As the PreRender event is recursively fired for all child controls, this event ensures the completion of the pre-rendering phase.
- **SaveStateComplete** - State of control on the page is saved. Personalization, control state and view state information are saved. The HTML mark-up is generated. This

stage can be handled by overriding the Render method or creating a Page_Render handler.

- **UnLoad** - The UnLoad phase is the last phase of the page life cycle. It raises the **UnLoad event for all controls recursively and lastly for the page itself**. Final clean-up is done and all resources and references, such as database connections, are freed. This event can be handled by modifying the OnUnLoad method or creating a Page_UnLoad handler.

1.3 Application Configuration:

In ASP.NET, there are **two files** are there to configure ASP.NET website. The **first file** is called **global.asax** file which **manages various events of ASP.NET website** and **another file called web.config file** which stores various settings applied on the website.

[1] global.asax file:

- The file 'global.asax' allows web developer to write the code, which will be executed on some **specific events** like, Application_Start, Application_End, Application_Error, Session_Start, Session_End and so on.
- To add 'global.asax' file, you need to do Right click on the website name in the solution explorer, and choose 'Add New Item' option. From the 'Add New Item' diaig box you need to select 'Global Application Class' option and click on 'Add' button.
- There will be **only one global.asax** file is there **for each website**.
- The file 'global.asax' has to be in either C# or in VB language. The structure of the global.asax file is presented below:

```
<%@ Application Language="C#" %>

<script runat="server">

    void Application_Start(object sender, EventArgs e)
    {
        // Code that runs on application startup
    }

    void Application_End(object sender, EventArgs e)
    {
        // Code that runs on application shutdown
    }
}
```

```
void Application_Error(object sender, EventArgs e)
{
    // Code that runs when an unhandled error occurs
}

void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started
}

void Session_End(object sender, EventArgs e)
{
    // Code that runs when a session ends.

    // Note: The Session_End event is raised only when
    // the sessionstate mode
    // is set to InProc in the Web.config file. If
    // session mode is set to StateServer
    // or SQLServer, the event is not raised.
}
</script>
```

[2] web.config file:

- The file ‘web.config’ is used to store the settings of the website. For example, you can add namespace or assembly, to create profile of the user, to set the authorization rules etc. you need to modify web.config file.
- The file ‘web.config’ is available in the XML (Extensible Mark-up Lanagauge).
- There can be more than one web.config file is there in any web-site, but in one directory there will be only one web.config file will be there.
- When even you create a new website, then one web.config file will be automatically added in the Root directory of the website. The structure of the web.config file is presented below:

```
<?xml version="1.0"?>
```

```
<!--
```

For more information on how to configure your ASP.NET application, please visit

<http://go.microsoft.com/fwlink/?LinkId=169433>

-->

```
<configuration>
```

```
    <system.web>
```

```
        <compilation debug="false" targetFramework="4.0">
```

```
    />
```

```
    </system.web>
```

```
</configuration>
```

1.4 Common Properties:

1. **ID:** ID property takes alphanumeric strings, to provide unique identification to the control on a webpage. Every control in a webpage has unique ID.
2. **RunAt:** Attribute RunAt indicates, who will process this control, server or client? For all ASP.NET controls, RunAt attribute is set to “server”, as the ASP.NET controls are processed by server.
3. **BackColor:** Attribute BackColor specifies the background color of the control. It gets or sets color name or its Hex color code.
4. **ForeColor:** Attribute ForeColor is used to specify foreground (font color) to the control. It gets or sets color name or its Hex color code.
5. **BorderWidth:** Attribute BorderWidth is used to specify width of the border surrounding to the control. It gets or sets integer value. For example, if you set value ‘2’ to this attribute, you will get a border of width 2-pixel surrounding to that control.
6. **BorderStyle:** The attribute BorderStyle is used to specify style of the border. Possible values to this attribute are None, Solid, Dashed, Dotted, Double, Groove etc.
7. **BorderColor:** The attribute called BorderColor is used to specify color of the border line surrounding the specific control. BorderColor attribute gets or sets the color name or Hex color code.
8. **CSSClass:** The attribute CSSClass is used to apply cascading style sheet class to the control. It takes CSS class name, and whatever formatting is specified within the class will be applied to that control.
9. **TabIndex:** The attribute TabIndex is used to set index number, which is used to focus when user presses Tab. It gets or sets integer number. When the page is loaded, control

with TabIndex 0 will be highlighted, if user presses Tab button, then a control having TabIndex 1 gets highlighted and so on.

10. **Font:** This attribute is used to set Font for the particular control. You can specify Font name, size, bold, italic, underline etc. It gets or sets object of font class.
11. **Enabled:** The attribute Enabled is used to Enable or Disable the control. It gets or sets Boolean value. If Enable attribute is set to TRUE, then user can interact with the specific control at runtime. If Enable attribute is set to FALSE, then user cannot interact with the control at runtime.
12. **ToolTip:** This attribute is used to set ToolTip for the specific control. It takes string, and at runtime when mouse pointer hover, over the control string written in the ToolTip attributes will be appeared in a small pop-up.
13. **Text:** Text property takes string from the user. String written in the Text property will be appear on the specific control.

1.5 Label Control:

Whenever you want to modify text, displayed in the page dynamically you can use Label.

For example:

```
Label1.Text= DateTime.Now.ToString("T")
```

This line on a form load event will change the display of the text at run time, when form loads. The additional properties of the Label control are shown below.

1. **EnableViewState:** Attribute EnableViewState takes Boolean value. By default, almost every control has this property and that is having 'TRUE' value. If you set, 'FALSE' value to this property, then browser will not send the information about specific control to the server. In that case server will fetch the information about this control from the default property set.
2. **AssociatedControlID:** Attribute AssociatedControlID is used to associate label with other controls. It takes ID of the other control within the page. Label control is rendered as a tag if the AssociatedControlID property is not set. If you are setting this property then Label control is rendered in the <label> tag. It is used for screen reader type of devices. Side benefit of using this property is by clicking on the label focus will be automatically set to associated control.

1.6 Textbox Control:

Textbox control is used to take the data from the user in string format. Textbox takes data inputted by the user. Following additional and important properties are the of the Textbox control.

1. **AccessKey:** It is used to provide access key to the Textbox control. It takes a character, which will be used as an access key. If you set AccessKey to 'N' for any textbox then, at runtime when user press Alt + N, then the focus will be shifted to that textbox.
2. **AutoCompleteType:** This attribute provides a set of classes like FirstName, LastName, Email, Cellular etc. The data imputed by the user in the textbox with specific class will be recorded at the client side, and will be retrieved automatically when user input the text into the textbox sharing similar class type on keypress. You can press Enter to complete the details and do not have to repeat the same text again.
3. **AutoPostBack:** This attribute takes Boolean value from the user. The default value for this attribute is FALSE. If we set this attribute of the textbox to TRUE, then the page will be automatically posted to the server when focus comes out of the textbox.
4. **ReadOnly:** This attribute accepts Boolean value and the default value of this attribute is FALSE. If the value for this attribute is set to TRUE, then user cannot make any changes into the Textbox, but yes user can see the text written in the textbox.
5. **MaxLength:** The attribute MaxLength of the textbox control takes integer value from the user. Default value for this attribute is 0, which means user can enter maximum 32768 characters into the textbox. If you set the value 10 to MaxLength attribute then user cannot enter more than 10 characters into the textbox.
6. **TextMode:** The possible values for this attribute are SingleLine, MultiLine, Date, DateTime, Email, Password etc. It is used to take multiline input, password, date, valid email address etc. The default value for this attribute is SingleLine.

1.7 Literal Control:

Literal Control is similar to the Label control. You can use the Literal control to display text or HTML content in a browser. However, unlike label control, the Literal control is not rendered as a tag. It is a very light weight control does not have any formatting features like BackColor, ForeColor, Borderwidth, BorderStyle and so on. Literal control has limited properties like ID, Text, EnableViewState and Mode.

Mode: The Mode property has default value "Transform". Other possible values are Encode and PassThrough. If you set any HTML tag in the text property of the Literal control and Mode is Encode then that text value will be encoded in such a way that, the HTML tag mention in the text property will be displayed as it is.

1.8 Panel control:

The panel control act as a container control, which contains other controls like Label, Textbox, RadioButton, CheckBox etc. Panel control is mainly used to group other controls which are placed inside the Panel control. If you set Enabled property of the Panel control is FALSE then all other control which are placed inside the Panel control will become non-operative of the user. Similarly, if you set Visible property of the

Panel control is FALSE, then all controls placed inside the panel control will not be visible to the user at runtime.

The other use of the Panel control is to generate other controls at runtime. The following code on the click event of the button will add a Textbox control within the Panel control at runtime.

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label l1 = new Label();
    TextBox t1 = new TextBox();
    l1.Text = "Enter Your Name:";
    Panel1.Controls.Add(l1);
    Panel1.Controls.Add(t1);
}
```



Exercises**Question: 1 Answer in short:**

1. What is ASP?
2. Why HTTP is called stateless protocol?
3. List all features of ASP.NET
4. List all features of ASP.NET IDE.
5. Explain Literal control

Question: 2 Explain following properties:

1. AutoPostBack (TextBox)
2. MaxLength (TextBox)
3. TextMode (TextBox)
4. Mode (Literal)
5. EnableViewState (Label)

Question: 3 Answer in brief:

1. What is ASP.NET? Explain execution model of ASP.NET webpages.
2. Explain the life cycle of ASP.NET webpage.
3. Explain Global.asax and web.config file in brief.
4. Explain built-in directory structure of ASP.NET
5. Write a short note on coding model of ASP.NET
6. What is code render block? Explain it with an example.
7. Explain the use of Panel control.

Question: 4 Answer the following MCQs:

Q:1 Web user control file is saved with _____ extension.

- | | |
|-------------|-----------|
| (A) .aspx | (B) .asmx |
| (C) .master | (D) .ascx |

Q:2 _____ is a light weight control, similar to label control with no formatting features.

- | | |
|----------------|--------------|
| (A) Textbox | (B) Calendar |
| (C) FileUpload | (D) Literal |

Q:3 _____ ASP.NET Build-in directory is used to pre-compiled store class files.

Q:4 In _____ coding model two separate files, one for coding and another for HTML designing is maintained.

- (A) SeparateFile coding model (B) Inline coding model
 (C) Code-behind model (D) HTML_VB coding model

Q:5 _____ web server is used to run ASP.NET web pages.

Q:6 .NET Framework consists of CLR and

- (A) FCL (B) CLF
(C) MSIL (D) IL

Q.7 IDE stand for

- (A) Information Data Encryption
 - (B) Integrated Development Environment
 - (C) Information Data Enterprise
 - (D) Internet Data Encryption system

Answers

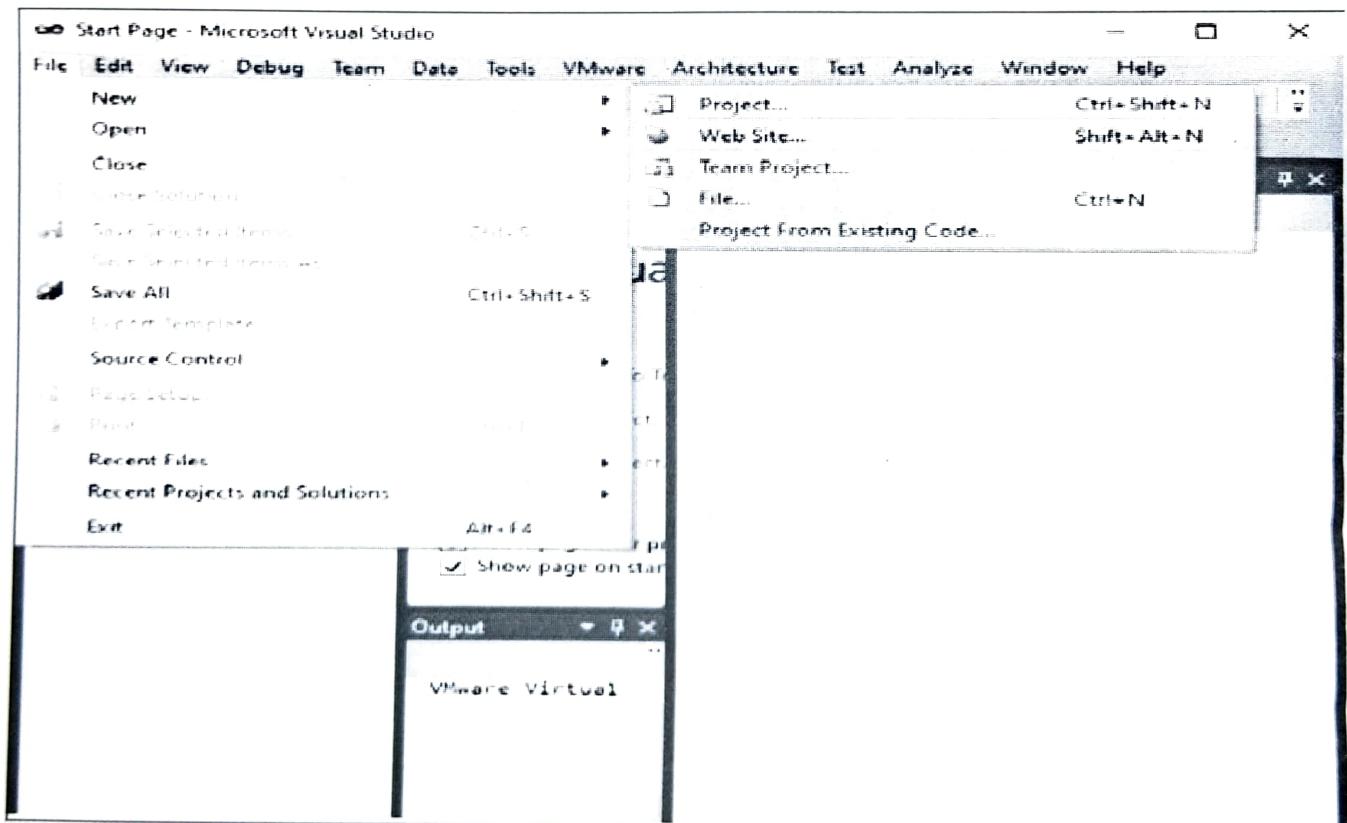
1. D [.ascx]
 2. D [Literal]
 3. C [Bin]
 4. C [Code-behind model]
 5. A [IIS]
 6. A [FCL] framework class library
 7. B [Integrated Development Environment]



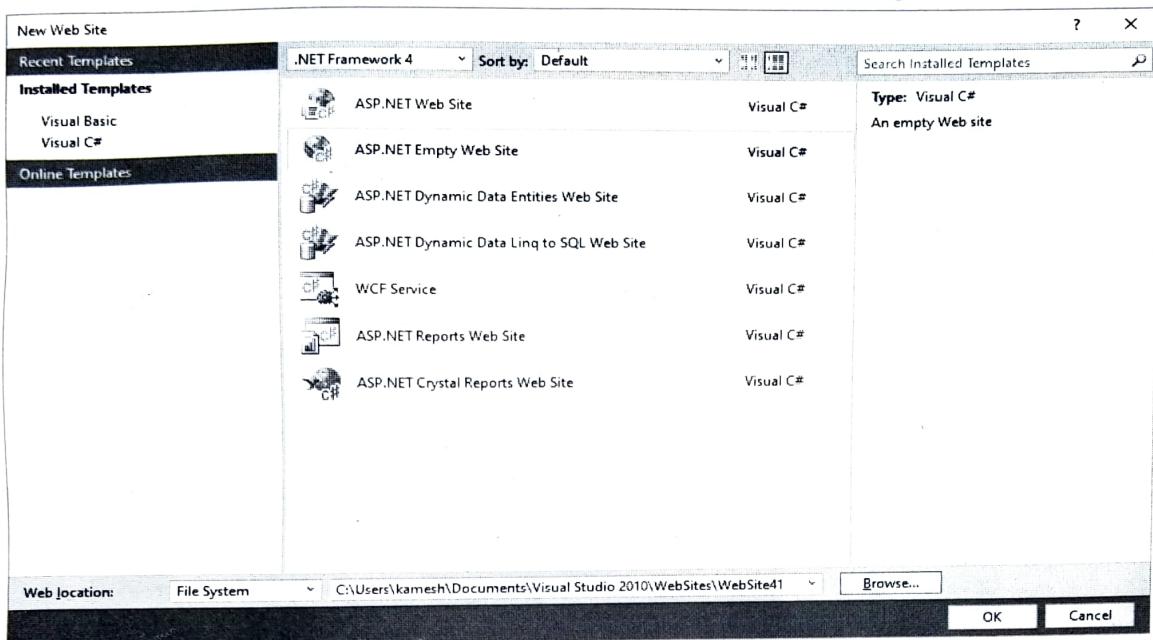
CC-304 Unit - 1 Web Application Development-I Practicals**Program : 1**

Design .aspx page, having 4 Textboxes (First name, Last name, Email and Mobile). Place a button on the page. On the click even of the button user will be redirected on another page, having same 4 Textboxes having AutoComplete capability. On another page user do not have type First name, Last name, Email, and Mobile number but it will be AutoComplete by pressing one or two keys in each textbox. (Demo of AutoCompleteType property).

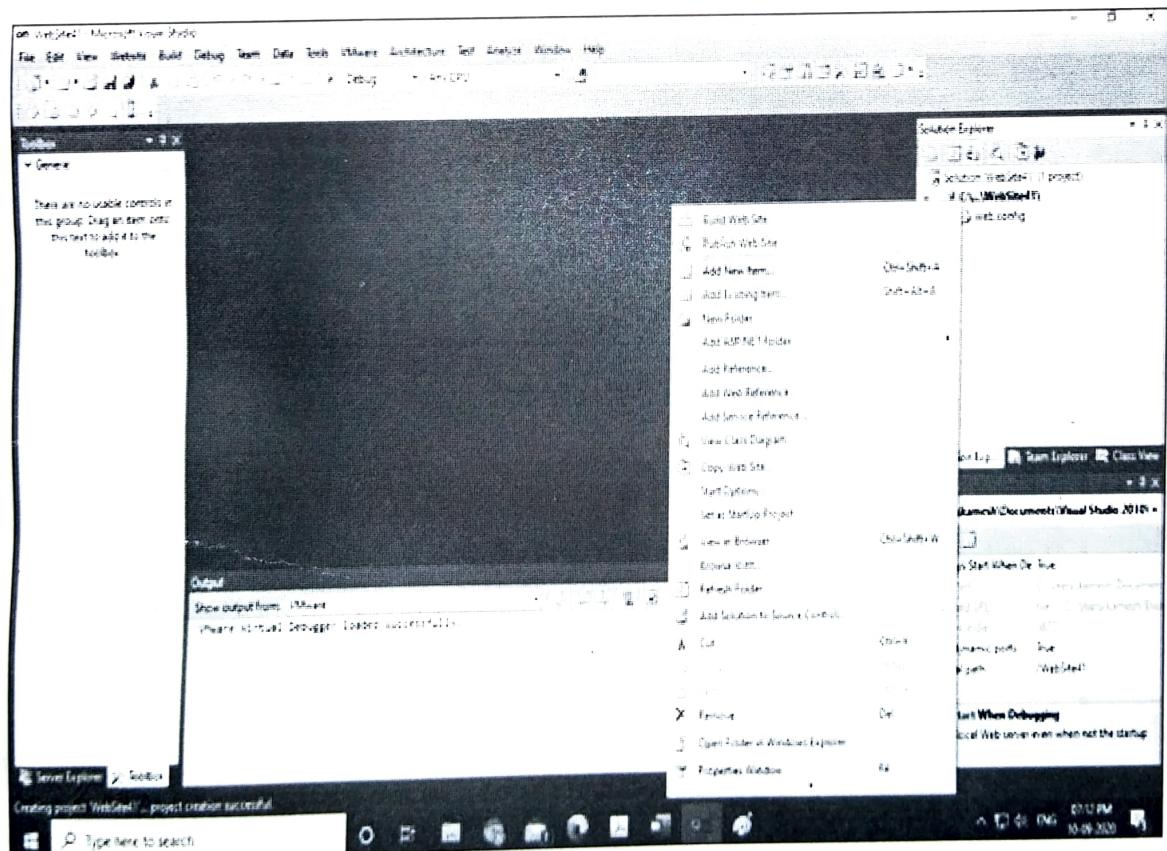
- Open the Visual Studio, Click on File Menu, Select New -> Website option as shown in the following figure.



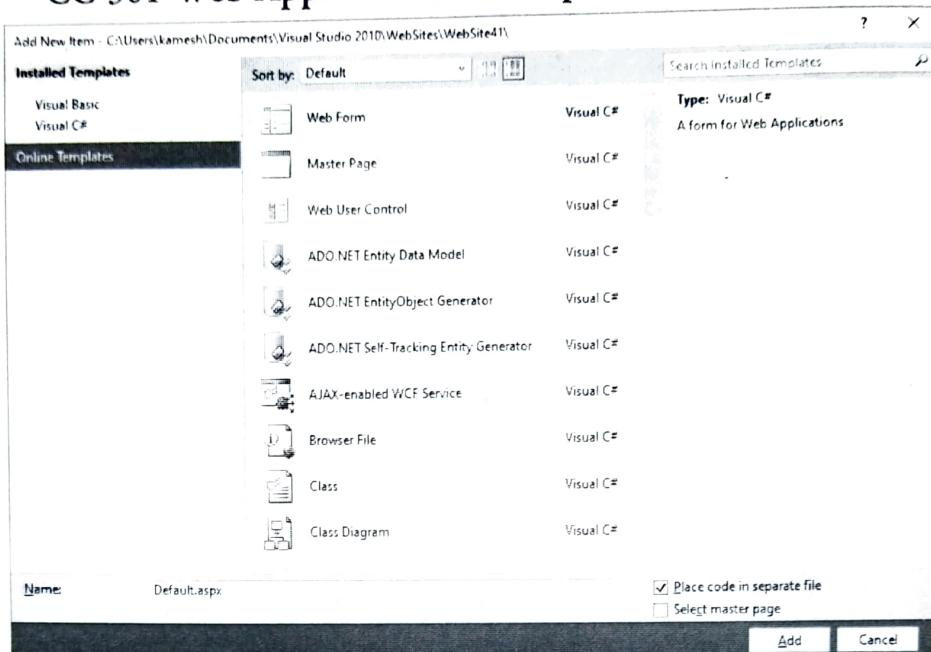
- Now from the dialog box, Select C#.NET from the Language option and select “ASP.NET Empty Website” from the Template. Click on OK Button.



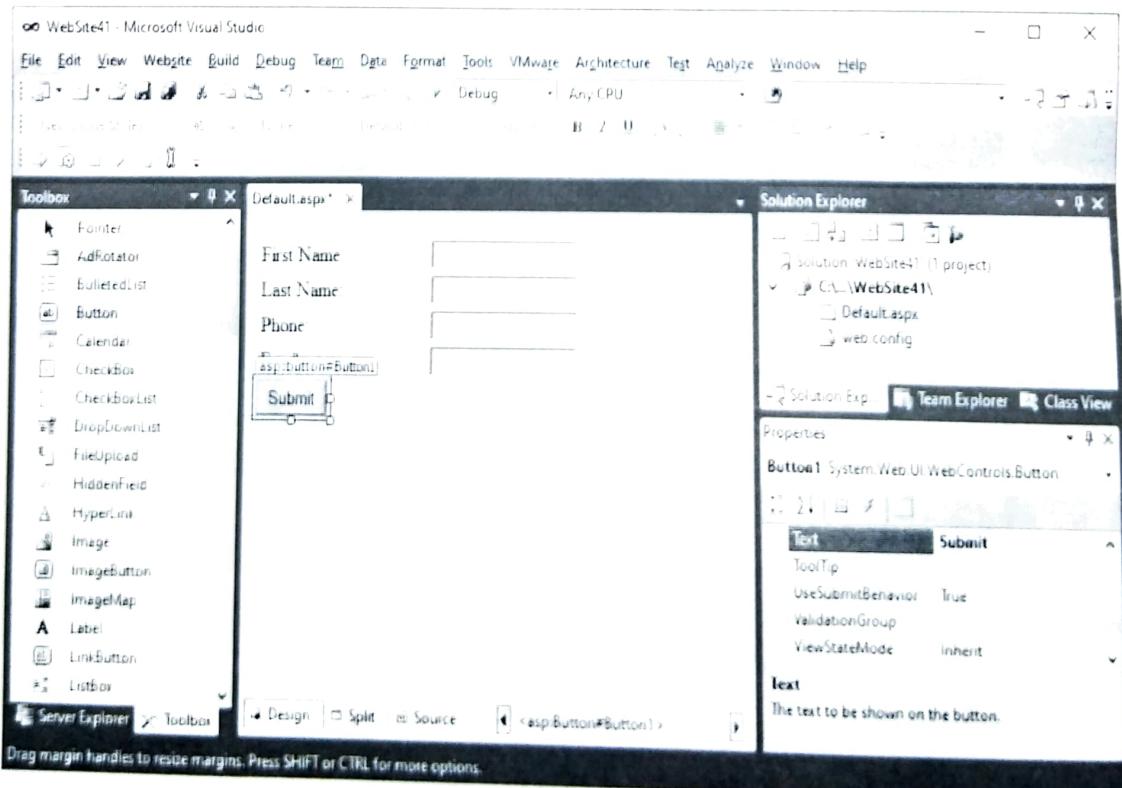
- Now Right Click on the WebSite in the Solution Explorer and choose option “Add New Item” as shown in the following figure.



- Now from the dialog box, select option Web Form and click on “Add” button. This action will add one web page called ‘default.aspx’ to your web site.

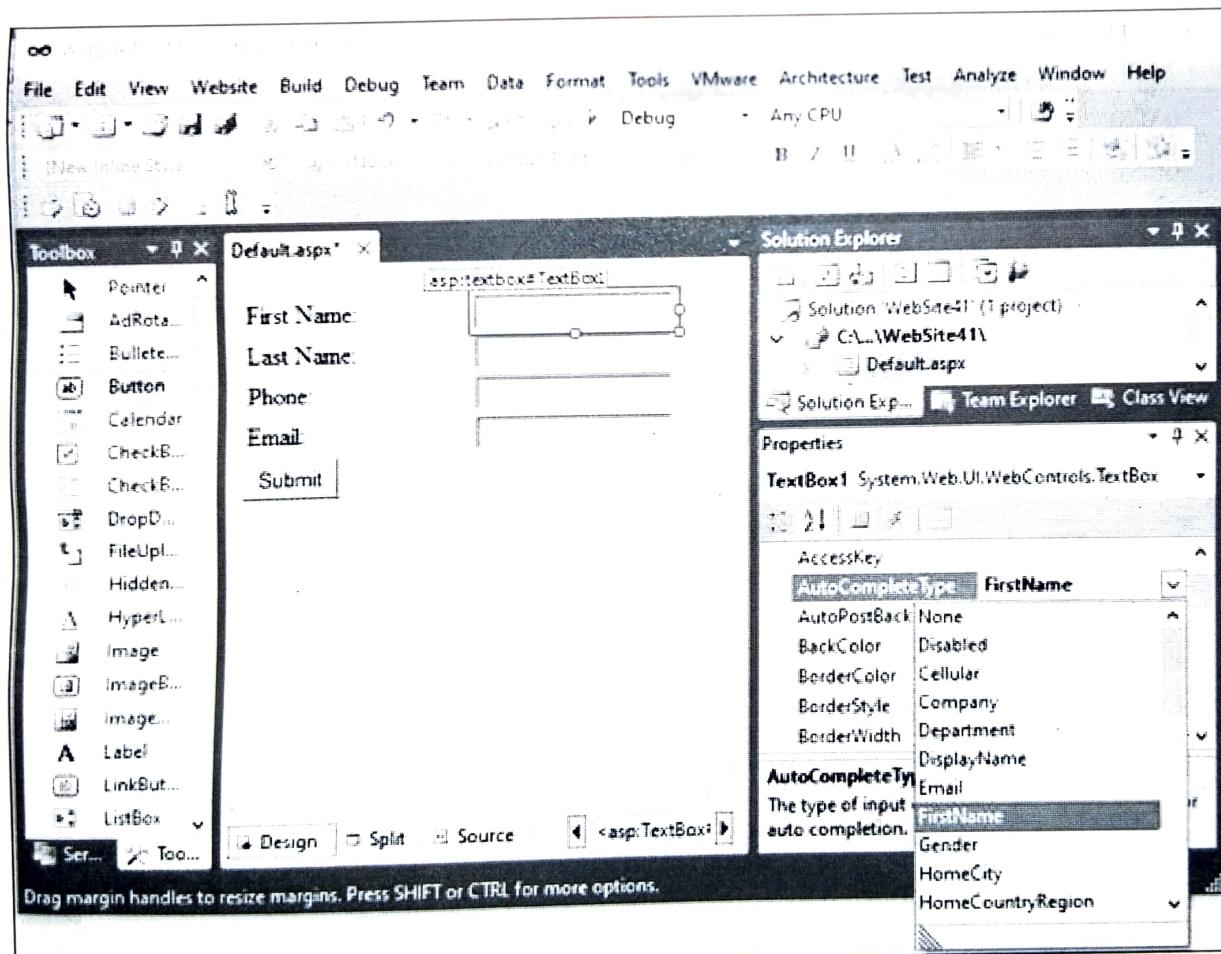


- To open the Design mode of the webpage, click on the Design tab (See at the bottom side three options are available, ‘Design’, ‘Split’, ‘Source’. Now click on the Tabl menu, select option “Insert Table”, Enter 4 in Rows and 2 in Columns and click on OK button. This will add a table with 4 Rows and 2 Columns in to your web page.
- Place 4 Textboxes in each row of the table in the 2nd column.
- Type “FirstName”, “LastName”, “Phone” and “Email” in each row of the 1st column.
- Place a button on the bottom of that table. Select the button and change its Text property to “Submit” as shown in the figure.



- Now select TextBox1 and set the AutoCompleteType property to “FirstName”, select TextBox2 and set the AutoCompleteType property to “LastName”, select TextBox3 and set the AutoCompleteType property to “Cellular”, and finally set the AutoCompleteType property to “Email” for TextBox4 as shown in the following figure.
- Add another web page by doing right click on the website (in the solution explorer), choosing ‘Add New Item’ option and by clicking on the OK button. This will add another page to your website called ‘Default2.aspx’.
- Design the Default2.aspx in the same as you have designed Default.aspx by taking 4 textboxes. Make sure in this web page you do not have to add button. Set the AutoCompleteType property of all 4 textboxes to “FirstName”, “LastName”, “Cellular” and “Email”.
- Now double click on the button we have placed on the Default.aspx web page, and add the following code on the click event of the button.

```
Response.Redirect("Default1.aspx");
```

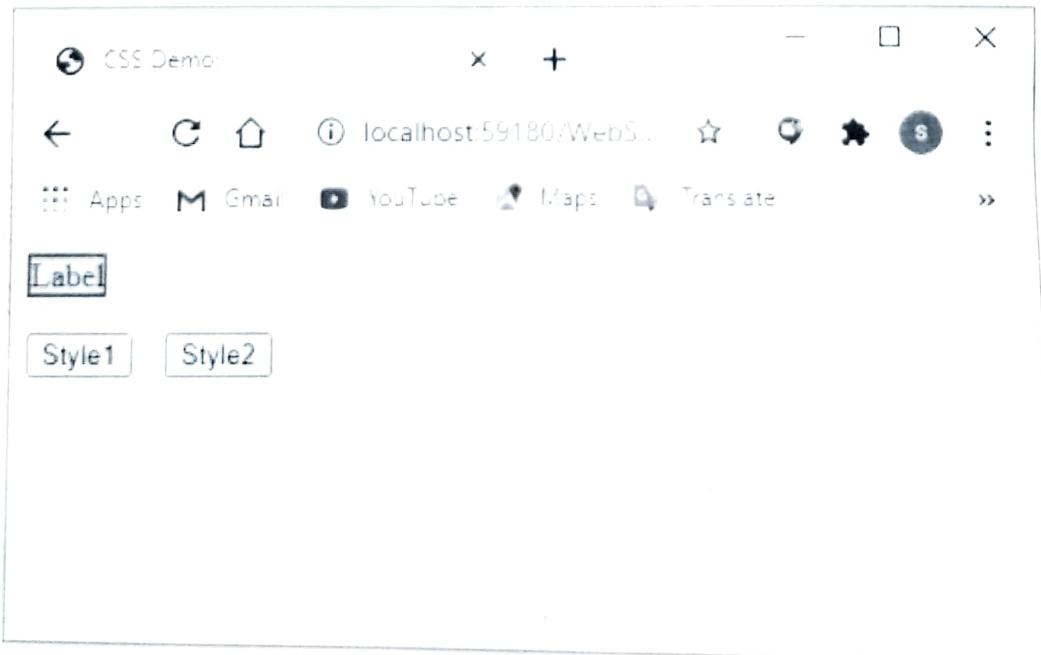


- Now, run the default.aspx in the web browser (by pressing Ctrl + F5), add the details such as FirstName, LastName, your mobile number and Email address. Click on the Submit button.
- Browser will navigate you to Default2.aspx. Now try to type same FirstName, LastName, MobileNo and Email. You will observe that when you enter first character the data you have entered in the default.aspx will be appeared in the small popup. You can press enter and the details will be automatically filled in the textbox.

Program : 2

Design two different CSS class in the web page having different formatting features like border size, border style, border color, font color, background color etc. Place two buttons and a label on the .aspx page. On the click event of the first button one CSS class will be applied to the label and on the click event of the second button apply second CSS class to the label. (Changing appearance of the label at run time using CSSClass property).

- Create a new empty website in the Visual Studio and add Default.aspx page to the website. By default, the page will be opened in the ‘Source’ mode.
- Create 2 style sheet classes in the head section as shown below. Click on the ‘Design’ to change the mode of a web page from ‘Source’ to ‘Design’.
- Place a label and 2 buttons on the web page. Change the text property of both the buttons to ‘Style1’ and ‘Style2’ as shown in the figure.



- The source code of the web page is given below:

CC-301 Web Application Development- I (Using C#)

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>CSS Demo:</title>
    <style type="text/css">
        .Style1
        {
            background-color : Yellow;
            color: Blue;
            border: 2px solid red;
        }
        .Style2
        {
            background-color : Aqua;
            color: Blue;
            border: 2px Dashed Black;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
            <br />
            <br />
        </div>
    </form>
</body>
```

```
<asp:Button ID="Button1" runat="server"
onclick="Button1_Click" Text="Style1" />
 &nbsp;&nbsp;
<asp:Button ID="Button2" runat="server"
onclick="Button2_Click" Text="Style2" />

</div>
</form>
</body>
</html>
```

- Open the web page in the design mode, double click on the Button1. It will open the procedure called “Button1_Click”. On the Click event of the “Button” write the following code.

Label1.CssClass = "Style1";

- In the same way on the click event of the button2 add the following code.

Label2.CssClass = "Style2";

- After writing the code, your default.aspx.cs file will be:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.CssClass = "Style1";
    }

    protected void Button2_Click(object sender, EventArgs e)
```

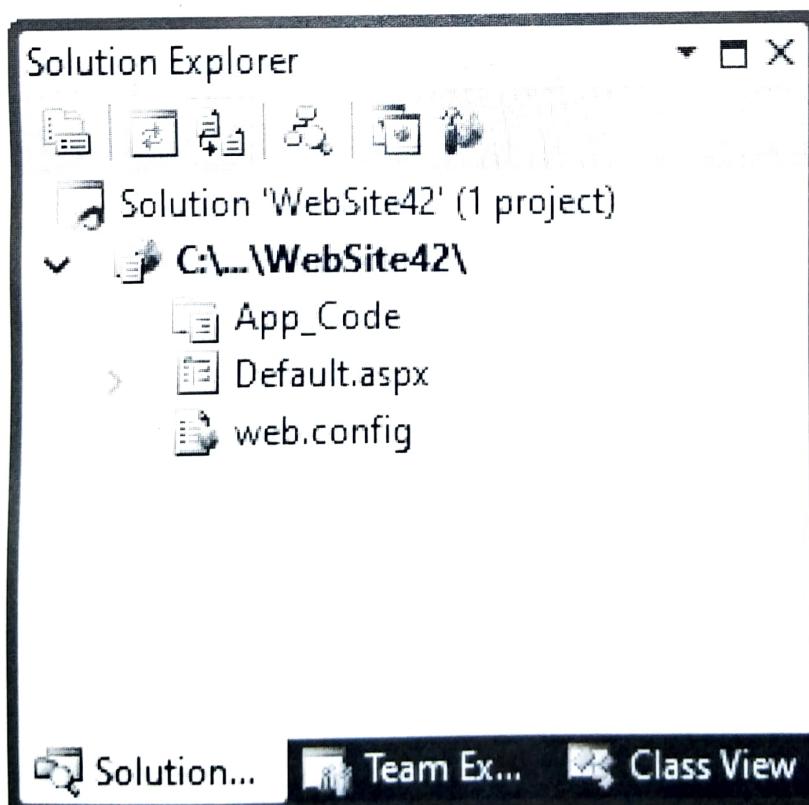
```
{  
    Label1.CssClass = "Style2";  
}  
}
```

Run the web page by pressing Ctrl+F5 key and click on the button1. It will apply the style1 to the Label. Now click on the Button2 it will apply style2 to the Label.

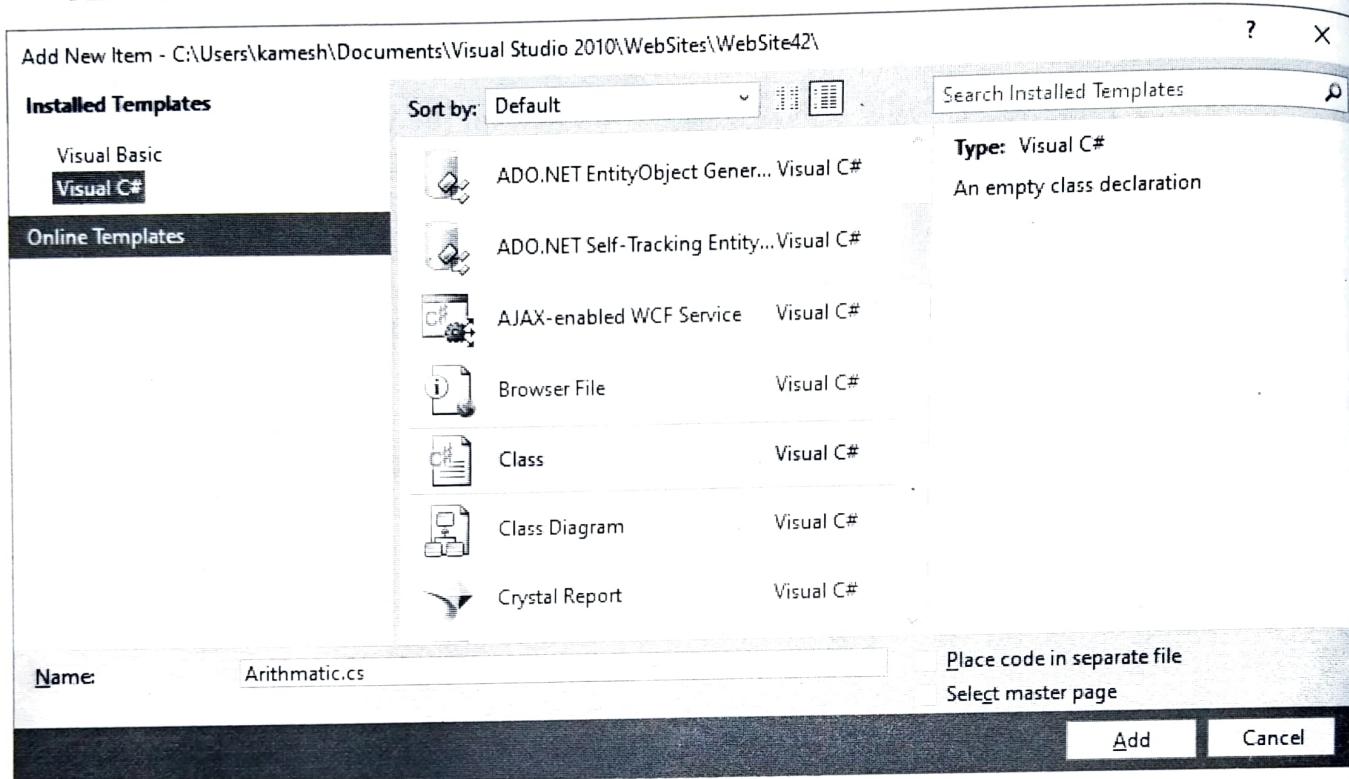
Program : 3

Design a class file having two methods to do sum and multiplication, which takes two arguments. Design a web page having two textboxes to take integer number from the user. Place two buttons to invoke sum and multiplication method. Print the resultant value in the label control placed on the web page. (Use of App_code directory).

- Create a new web site using Visual Studio, and Add a default.aspx page to the web site. Now Right click on the website in the Solution Explorer and Choose option: Add ASP.NET Folder -> App_Code.
- This will add a directory App_Code to your website. Usually this directory is used to store the (code) class files of the website.



- Now Right Click on the App_Code directory and choose option “Add New Item”. This action will open the dialog box. From the “Add New Item” dialog box, choose option Class. Give a proper name to your class file. Here in the example we have given “Arithmatic.cs” class.



- This action will add the class “Arithmatic.cs” to your website in the App_Code directory. Now create two methods called ‘Sum’ and ‘Multy’ in this class file as shown below.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for Arithmatic
/// </summary>
public class Arithmatic
{
    public Arithmatic()
    {
        //
```

```

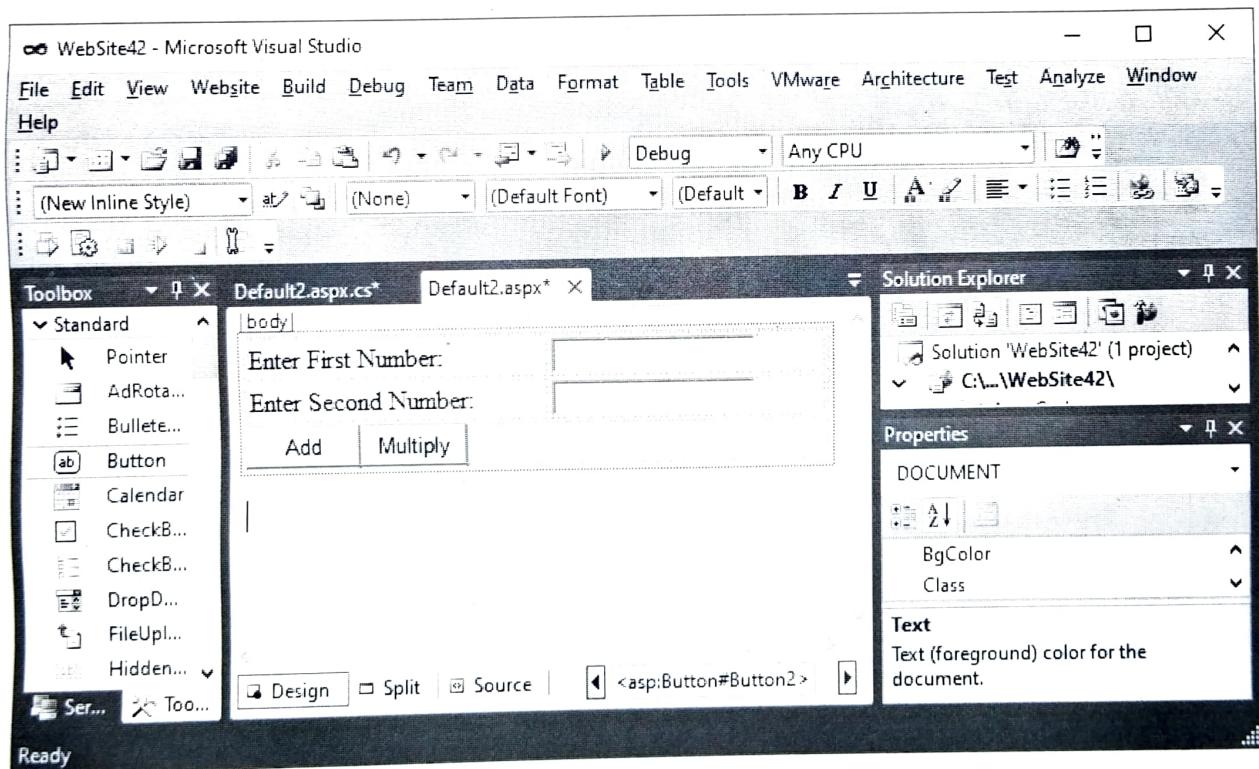
// TODO: Add constructor logic here
//
}

public int sum(int Num1, int Num2)
{
    return (Num1 + Num2);
}

public int multy(int Num1, int Num2)
{
    return (Num1 * Num2);
}

```

- Now goto the default.aspx page and design the page by placing 2 Textboxes and 2 Buttons as shown below.



- Double click on the 'Add' button and write the following code on the click event of the button.

```

protected void Button1_Click(object sender, EventArgs e)
{
    Arithmetic obj = new Arithmetic();
    int ans;

```

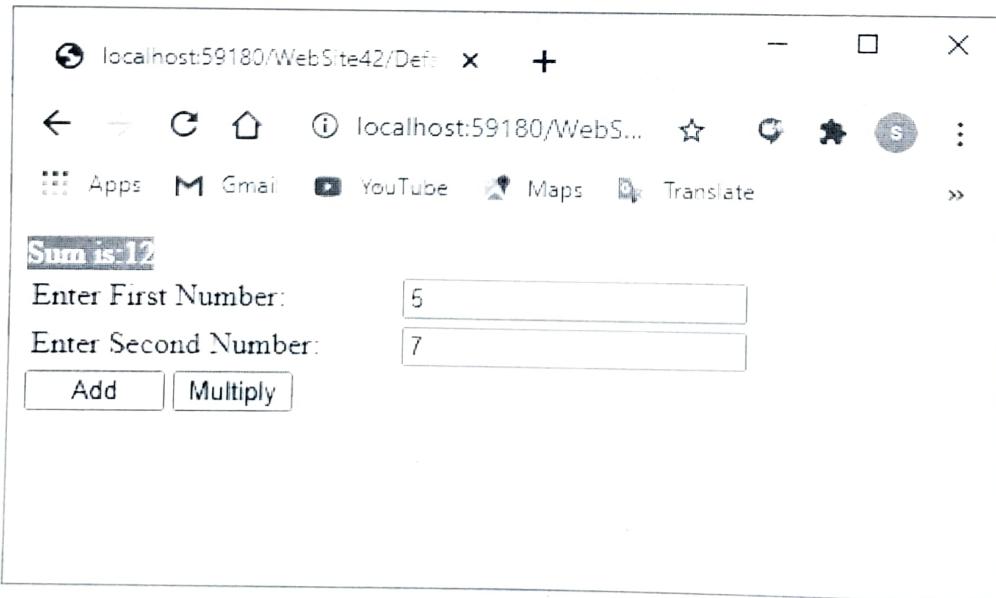
CC-301 Web Application Development- I (Using C#)

```
ans = obj.sum(Convert.ToInt16(textBox1.Text),  
Convert.ToInt16(textBox2.Text));  
Response.Write("Sum is:" + Convert.ToString(ans));  
}
```

- Similarly, double click on the button ‘Multiply’ button. On the click event of the ‘Multiply’ button write following code.

```
protected void Button2_Click(object sender, EventArgs e)  
{  
    Arithmetic obj = new Arithmetic();  
    int ans;  
    ans = obj.multy(Convert.ToInt16(textBox1.Text),  
    Convert.ToInt16(textBox2.Text));  
    Response.Write("Multiplication is:" +  
    Convert.ToString(ans));  
}
```

- Save and run the web page enter any two numbers in the Textboxes and click on the “Add” button. You will get the following output.

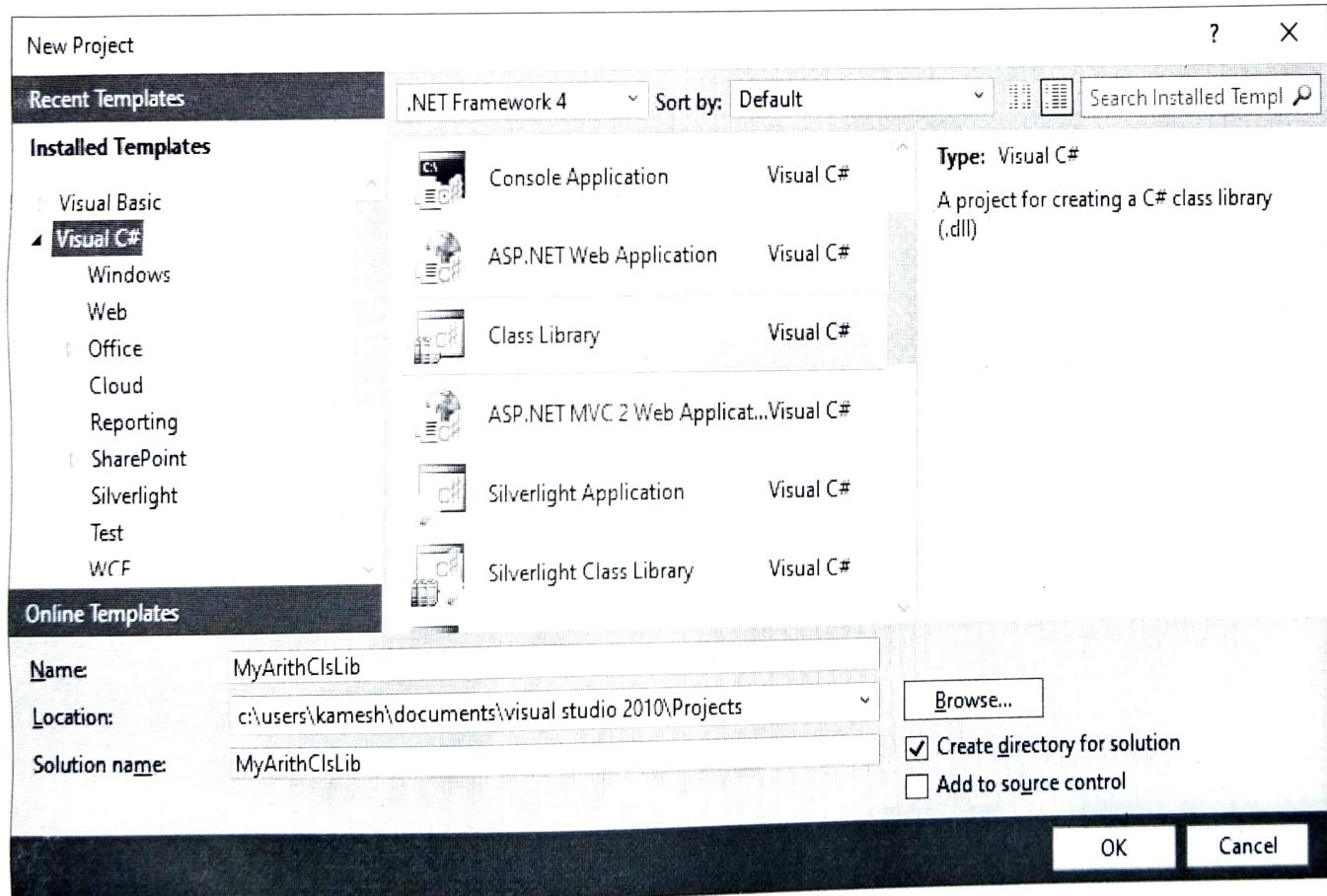


- Now click on the “Multiply” button. You will get multiplication of both numbers.

Program : 4

Create .dll class library file having 2 classes and each class has at least two methods. Add the .dll file into the ASP.NET website Design a web page to invoke the methods of .dll files. Use appropriate textboxes, label and button controls (Use of Bin directory).

- In this question first we need to create .dll file: So, first we will discuss how to create dll file. Dynamic Linked Library (.dll) file is a library (namespace) where one or more class files are there in the compiled form.
- In the previous question we have built a class file (.cs), which was not compiled file. Any person can open that class file into the Notepad kind of editor and can see the code written in that class file as well user can also do modification in it.
- Dynamic Linked Library (.dll) files are precompiled file, user can not see the coding which is written under it as well as user can not do any kind of modification in it. To build .dll file open Visual studio and select option File->New->Project.
- Now from the “New Project” dialog box, Select C#.NET language, “Class Library” project template. Give a name to your library to “MyArithClsLib” as shown in the following figure. Click on OK button.



- This will create a new Class Library which will have one default class that is Class1.cs.
- Now from the solution explorer, rename that class from “class1.cs” to “SimpleArith.cs”. It will show following message.

Microsoft Visual Studio



You are renaming a file. Would you also like to perform a rename in this project of all references to the code element 'Class1'?

- Click on Yes button. Now add two methods ‘Sum’ and ‘Subtract’ to this class file. Write the following code to the class file.

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

```
namespace MyArithClsLib
{
    public class SimpleArith
    {
        public int Sum(int Num1, int Num2)
        {
            return (Num1 + Num2);
        }
        public int Subtract(int Num1, int Num2)
        {
            return (Num1 - Num2);
        }
    }
}
```

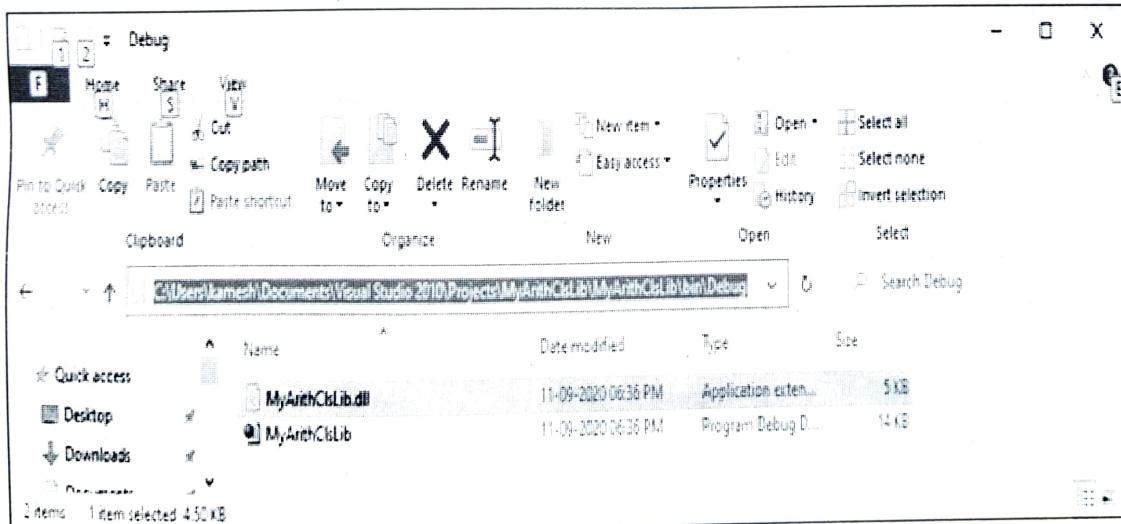
- Now do the rights click “MyArithClsLib” in the solution explorer, and choose option Add New Item. From the dialog box of “Add New Item”, select class and give class name “AdvArith.cs”. Click on the OK button.
- Now write the following code to the file “AdvArith.cs” class file.

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

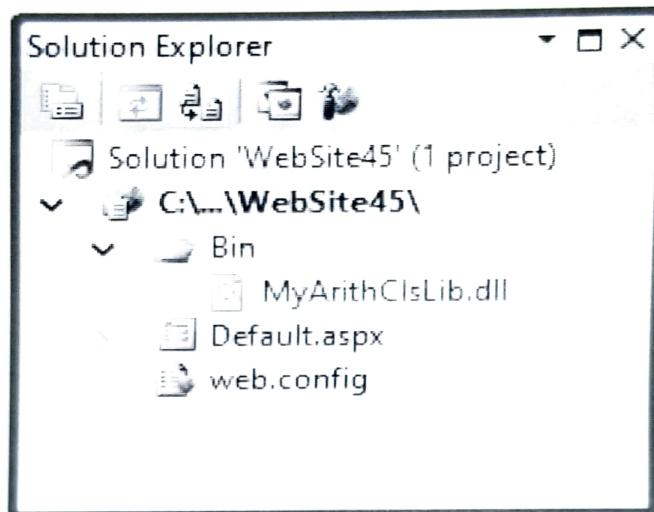
namespace MyArithClsLib
{
    public class AdvArith
    {
        public int Multy(int Num1, int Num2)
        {
            return (Num1 * Num2);
        }

        public int Divide(int Num1, int Num2)
        {
            return (Num1 / Num2);
        }
    }
}
```

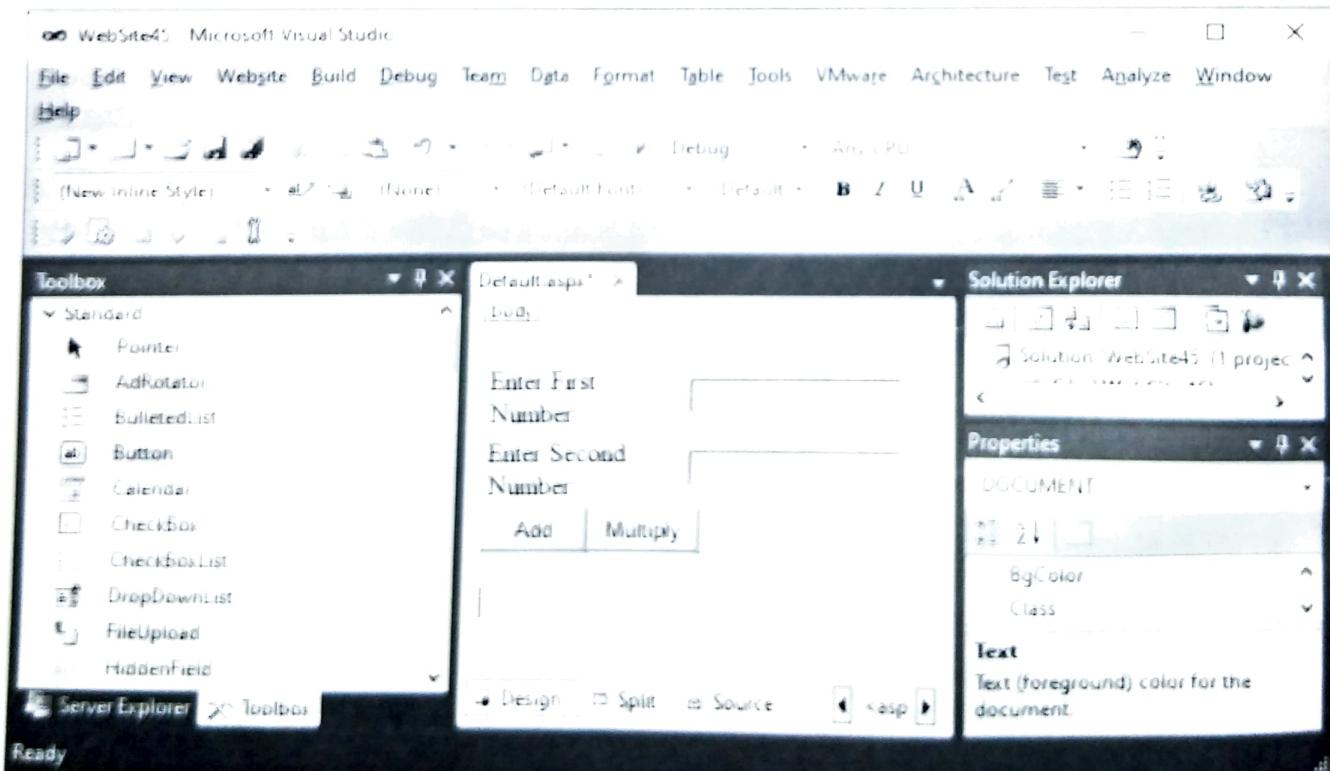
- Once the code is written click on Build menu and select option “Build Solution” (ctrl+Shift+B). This action will compile the Class Library and build the .dll file for you.
- Once you will get the message that “Build Succeeded”, close the project. Now from the My Computer you go to that location where you have created this project solution.



- If you go to ... \Projects\MyArithClsLib\MyArithClsLib\bin\Debug path you will be able to get MyArithClsLib.dll file. Copy this file.
- Now open the Visual Studio again and create a new website. Right click on the website and choose option “Add ASP.NET Directory”. From the list click on Bin. This will add a Bin directory to your solution explorer. Now Right Click on the Bin directory and paste that MyArithClsLib.dll file.



- Now add a web page called default.aspx to your website and design the web page exactly in the same way as we have designed the web page in the previous question. Take 2 Textboxes and 2 Buttons as shown in the following figure.



- Now add the following code on the click event of both buttons:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

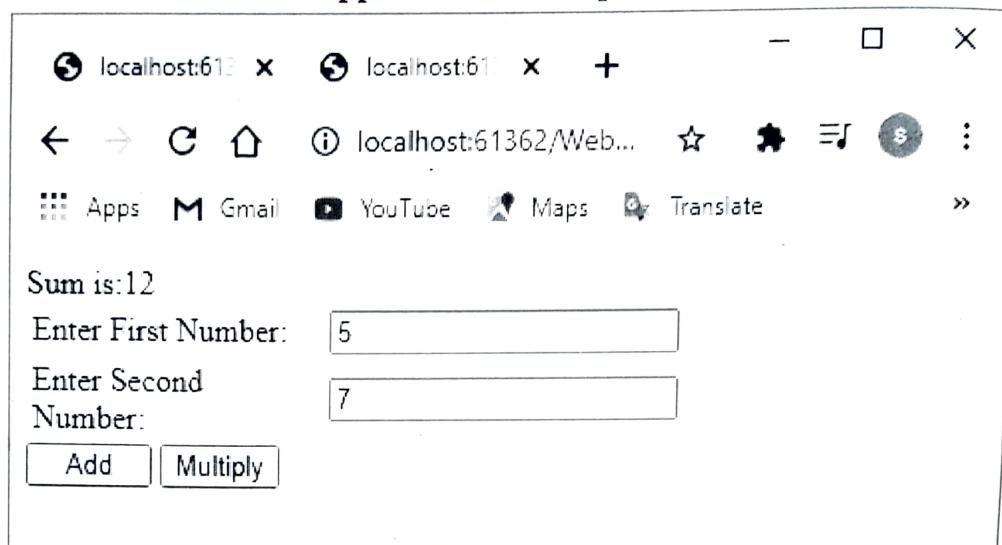
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        MyArithClsLib.SimpleArith obj = new
        MyArithClsLib.SimpleArith();
        int ans;
        ans = obj.Sum(Convert.ToInt16(TextBox1.Text),
        Convert.ToInt16(TextBox2.Text));
        Response.Write("Sum is:" + Convert.ToString(ans));
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        MyArithClsLib.AdvArith obj = new
        MyArithClsLib.AdvArith();
        int ans;
        ans = obj.Multy(Convert.ToInt16(TextBox1.Text),
        Convert.ToInt16(TextBox2.Text));
        Response.Write("Multiplication is:" +
        Convert.ToString(ans));
    }
}
```

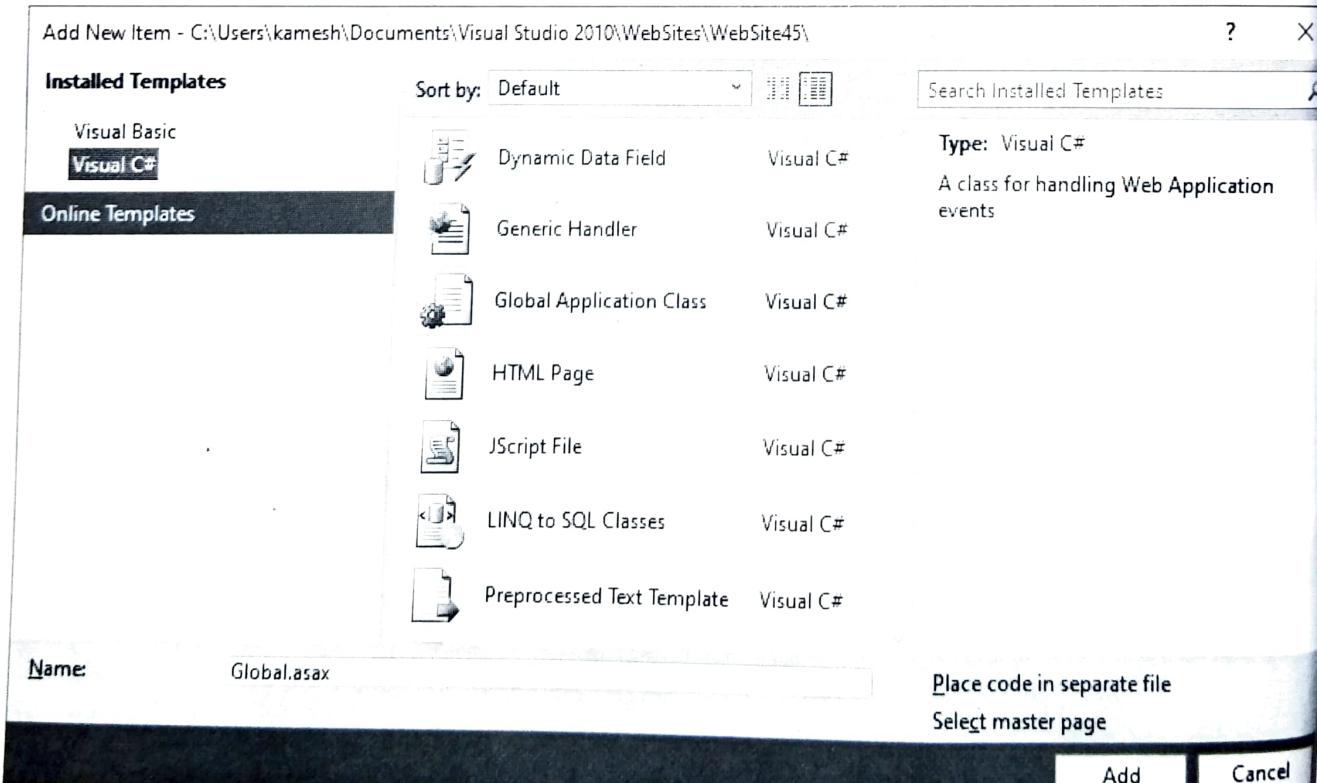
- Now run and test the web site.



Program : 5

Create a page which will show number of visitors of a page in label (using global.asax).

- Create a new “ASP.NET Empty WebSite” using Visual Studio. Now right click on the website in the solution explorer and choose option “Add New Item”.
- “Add New Item” dialog box will be appeared, and choose “Global Application Class” as shown in the following figure.



- Click on Add button, this will add a file called “Global.asax” to your website. This file contains various subroutines like “Application_Start”, “Application_Error”, “Application_End”, “Session_Start” and “Session_End”. Add the following code to this file.

```
<%@ Application Language="C#" %>
<script runat="server">

void Application_Start(object sender, EventArgs e)
{
    // Code that runs on application startup
    Application["users"] = 0;
}

void Application_End(object sender, EventArgs e)
{
    // Code that runs on application shutdown
}

void Application_Error(object sender, EventArgs e)
{
    // Code that runs when an unhandled error occurs
}

void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started
    Application["users"] =
        Convert.ToInt16(Application["users"]) + 1;
}

void Session_End(object sender, EventArgs e)
{
    // Code that runs when a session ends.

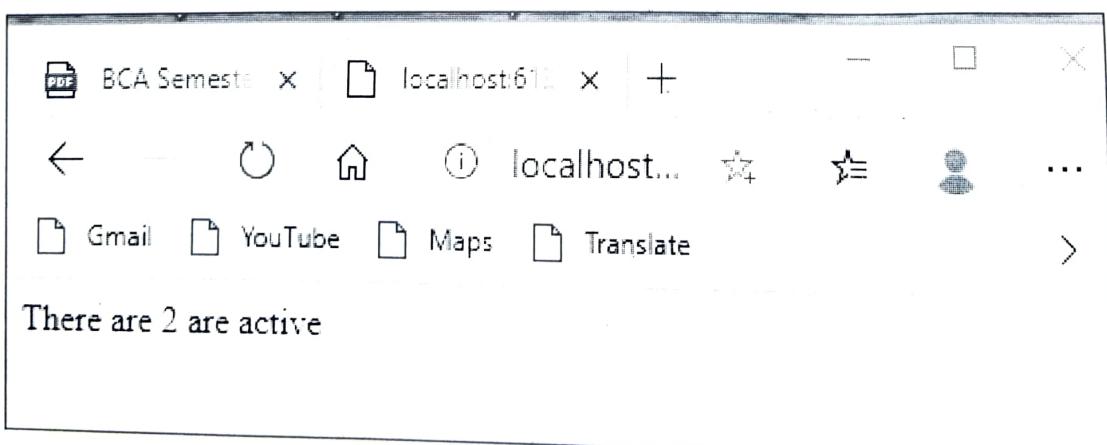
    // Note: The Session_End event is raised only when the
    // sessionstate mode
    // is set to InProc in the Web.config file. If session
    // mode is set to StateServer
    // or SQLServer, the event is not raised.
```

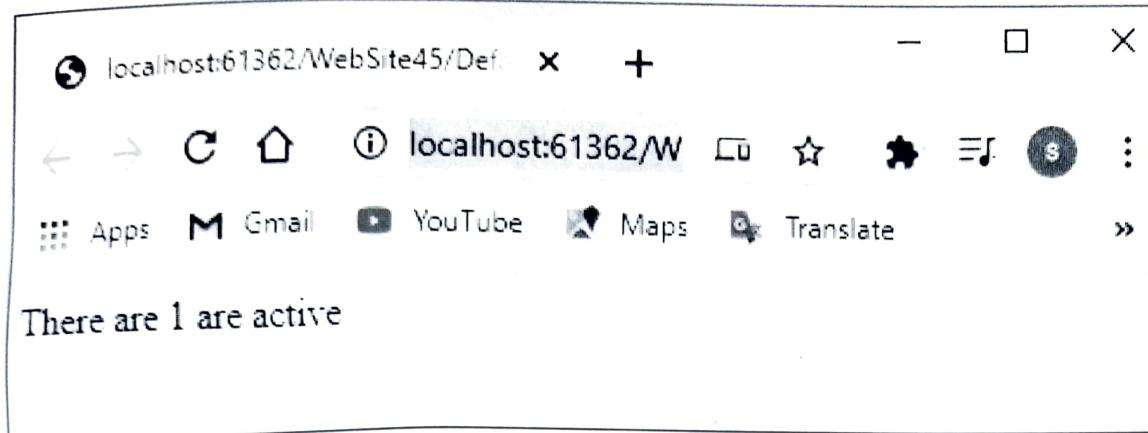
```
Application["users"] =  
Convert.ToInt16(Application["users"]) - 1;  
}  
</script>
```

- Now add a web page default.aspx to your website, by doing Right Click on the website from Solution Explorer, and by choosing “Add New Item” option. Place a label on the web page. Now double click on the web page it will open “Page_Load” subroutine. Write the following code to the “Page_Load” event of the default.aspx web page.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
  
public partial class Default : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
        Label1.Text = "There are " +  
Convert.ToString(Application["users"]) + " are active";  
    }  
}
```

- Run the web page into two separate browsers as shown and observe the output.





Program : 6

Design a webpage which has textbox and a button. User will enter his/her name in the textbox. On the click event of the button name of the user and current date time will be displayed on the title bar of the web page. (Using Literal control).

- Create a new “ASP.NET Empty Website” using Visual Studio and add default.aspx webpage to the website. Place a Literal control and Button by drag and drop to the “Source” of a web page as shown below.
- Make sure you need to put “Literal” control between <Title> and </Title> tag, in the head section of the web page. Place a button on the web page and change its Test Property to “Click Me and See in the Title bar”

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeFile="Default3.aspx.cs" Inherits="Default3" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
    <title>  
        <asp:Literal ID="Literal1" runat="server"  
        Text="Literal Demo"></asp:Literal>  
    </title>  
</head>
```

```
<body>
    <form id="form1" runat="server">
        <div>

            <asp:Button ID="Button1" runat="server"
Text="Click Me and See Title bar" />

        </div>
    </form>
</body>
</html>
```

- Add the following code on the click event of the button, as shown below:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default3 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Literal1.Text=DateTime.Now.ToString("T");
    }
}
```

- Run the application, and see in the tile bar of the browser. You will get the text "Literal Demo". Now click on the button you have placed on the web page. After a click again see the title bar of the browser. You will observe that instead of "Literal Demo", system date and time in the title bar of the browser.

Program : 7

Create an application Hit counter, which count the total number of users visited the page. (Using global.asax).

- Create a new “ASP.NET Empty website” using Visual Studio. Right Click on the website in the solution explorer and global.asax file (As we have included in the Program:5).
- Write the following code on the Application_Start event and Session_Start event in the global.asax file.

```
<%@ Application Language="C#" %>

<script runat="server">

void Application_Start(object sender, EventArgs e)
{
    // Code that runs on application startup
    Application["Cntr"] = 0;
}

void Application_End(object sender, EventArgs e)
{
    // Code that runs on application shutdown
}

void Application_Error(object sender, EventArgs e)
{
    // Code that runs when an unhandled error occurs
}

void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started
    Application["Cntr"] =
    Convert.ToInt16(Application["Cntr"]) + 1;
}
```

```
void Session_End(object sender, EventArgs e)
{
    // Code that runs when a session ends.

    // Note: The Session_End event is raised only when
    // the sessionstate mode
    // is set to InProc in the Web.config file. If
    // session mode is set to StateServer
    // or SQLServer, the event is not raised.

}
</script>
```

- Now, add a webpage to your website by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”. From the dialog box of “Add New Item” select “web form” and click on “Add” button.
- This action will add a defulat.aspx page to your website. Now click on “Design” (appeared in bottom left side) to change the mode from “Source” to “Design”. Place label control on the web page from the ToolBox.
- Now double click on the webpage, it will open “Form_Load” event. On the Form_Load event write the following code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

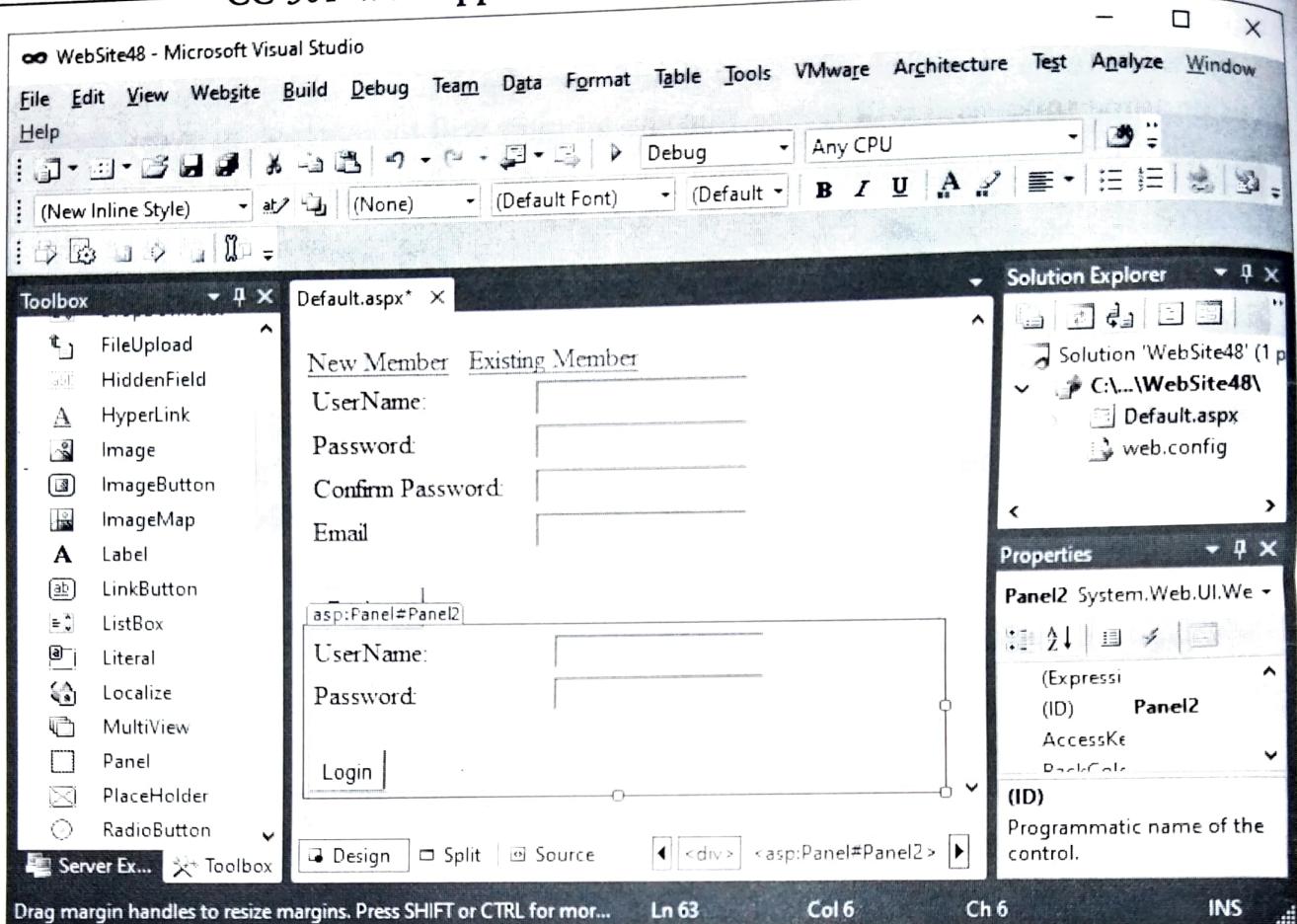
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = "This website has been visited for
+ "<B>" + Convert.ToString(Application["Cntr"]) + "</B>
times";
    }
}
```

- Run the default.aspx web page in the browser by pressing (Ctrl+F5) keys. You can see “This website has been visited for 1 times” message. If you close the webpage and run it after some time, you will notice that the counter will be increase in every fresh run of the web page.

Program : 8

Take to linkbuttons showing ‘New Member’ and ‘Existing Member’. When user clicks on the ‘New Member’ link button panel1 becomes visible, having user name, password, confirm password and email as inputs. When user clicks on ‘Existing Member’ link button then only panel2 becomes visible having user name and password as inputs). Set proper property of the textbox to mask the password.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button.
- Place two link buttons on the web page from the ToolBox. Change the Text property of the first LinkButton to ‘New Member’ and second LinkButton to ‘Existing Member’.
- Now, add panel control from the ToolBox and insert a table in the panel control using Table menu. Table should have 4 rows and two columns as shown in the figure. Write “UserName”, “Password”, “confirm Password” and “Email”, in 4 rows of the 1st column. Place 4 textboxes in each row of the table in the 2nd column.
- Select the textboxes we have taken to accept password and confirm password and change its TextMode property to “password” from the property window.
- After table, add a button control and change its Text property to “Register”.
- Now add another panel (panel2) to your web page, and inside this panel add table with 2 rows and 2 columns. In first column of the table you need to write “UserName” and “Password” in each row of the table. In the second column you need to place 2 textboxes one in each row. Select the textbox which we have taken to accept the password from the user, and set its TextMode property to “password” from the property window.
- Place one more button in the panel2, and change its Text property to “Login”. Set the visible property of “False” for both panel controls.



- This form design will generate following source code.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .style1
        {
            width: 100%;
        }
        .style3
```

```
{  
    width: 131px;  
}  
.style4  
{  
    width: 142px;  
}  
}  
</style>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
  
            <asp:LinkButton ID="LinkButton1"  
runat="server">New Member</asp:LinkButton>  
&nbsp;&nbsp;  
            <asp:LinkButton ID="LinkButton2"  
runat="server">Existing Member</asp:LinkButton>  
            <br />  
            <asp:Panel ID="Panel1" runat="server"  
Visible="False">  
                <table class="style1">  
                    <tr>  
                        <td class="style3">  
                            UserName:</td>  
                        <td>  
                            <asp:TextBox ID="TextBox1"  
runat="server"></asp:TextBox>  
                        </td>  
                    </tr>  
                    <tr>  
                        <td class="style3">  
                            Password:</td>  
                        <td>  
                    </td>
```

```
<asp:TextBox ID="TextBox2"
runat="server" TextMode="Password"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style3">
    Confirm Password:</td>
<td>
    <asp:TextBox ID="TextBox3"
runat="server" TextMode="Password"></asp:TextBox>
</td>
</tr>
<tr>
<td>Email</td>
<td>
    <asp:TextBox ID="TextBox6"
runat="server"></asp:TextBox></td>
</tr>
</table>
<br />
<asp:Button ID="Button1" runat="server"
Text="Register" />
</asp:Panel>
<asp:Panel ID="Panel2" runat="server"
Visible="False">
<table class="style1">
<tr>
<td class="style4">
    UserName:</td>
<td>
    <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
```

```
<td class="style4">
    Password:</td>
<td>
    <asp:TextBox ID="TextBox5"
runat="server" TextMode="Password"></asp:TextBox>
    </td>
</tr>
</table>
<br />
<asp:Button ID="Button2" runat="server"
Text="Login" />
</asp:Panel>
</div>
</form>
</body>
</html>
```

- Now go back to design mode, and double click on the LinkButton1 and write the code which is given below, similarly double click on the LinkButton2 and write following code on the click event of the linkbutton2.

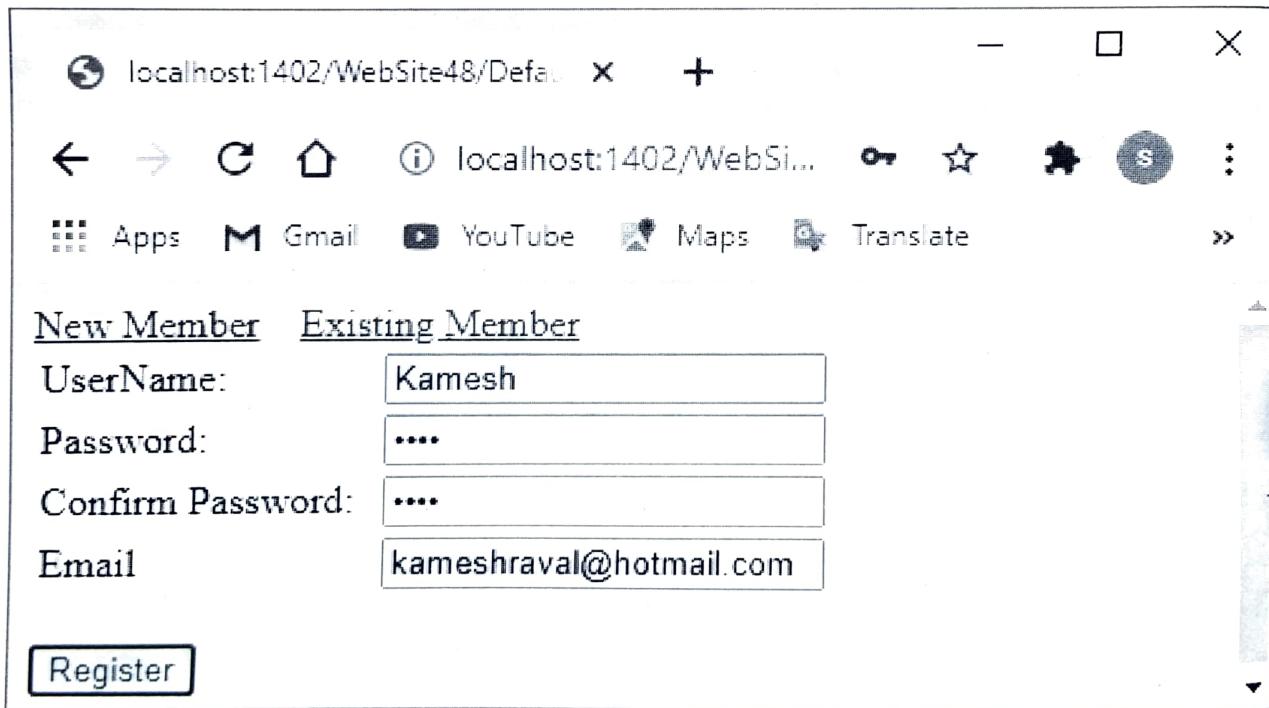
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

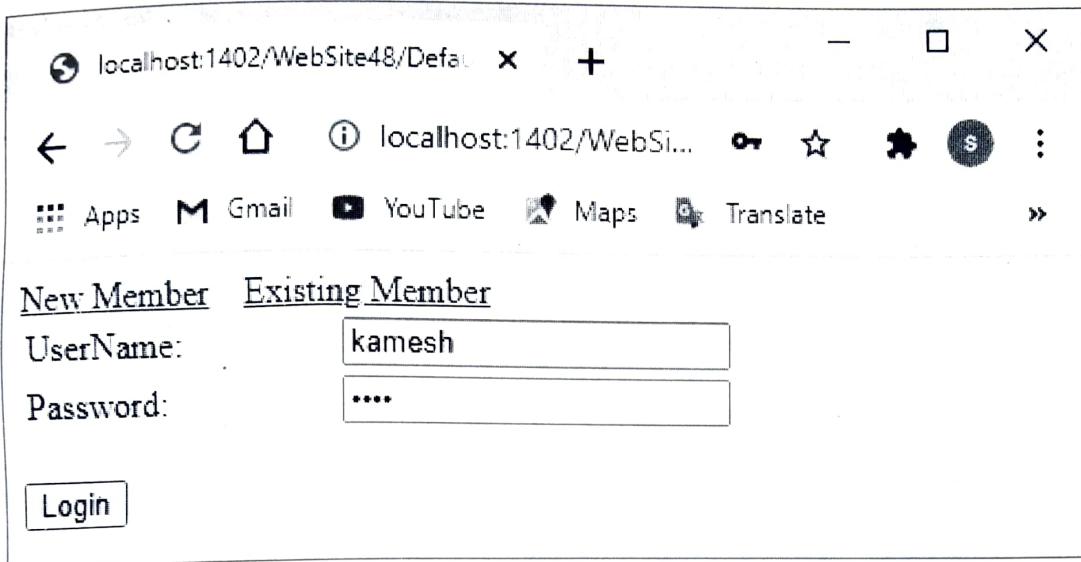
    }
    protected void LinkButton1_Click(object sender,
EventArgs e)
```

```
{  
    Panel1.Visible = true;  
    Panel2.Visible = false;  
}  
  
protected void LinkButton2_Click(object sender,  
EventArgs e)  
{  
    Panel1.Visible = false;  
    Panel2.Visible = true;  
}  
  
}
```

- Now run the web page in the browser by pressing Ctrl+F5.
You can see two link buttons “New Member” and “Existing Member”.
If you click on “New Member” then panel1 will be visible as shown below.



- Similarly, when you click on the “Existing Member” link button then panel2 will be visible as shown below.



Program : 9

Design .aspx web page which prints “Gujarat University” for 5 times, each in a new row with increasing font size by 1 each time. (Use loop in c# using code render block).

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button.
- Now write the following code to the source code, run and view your webpage in the browser.

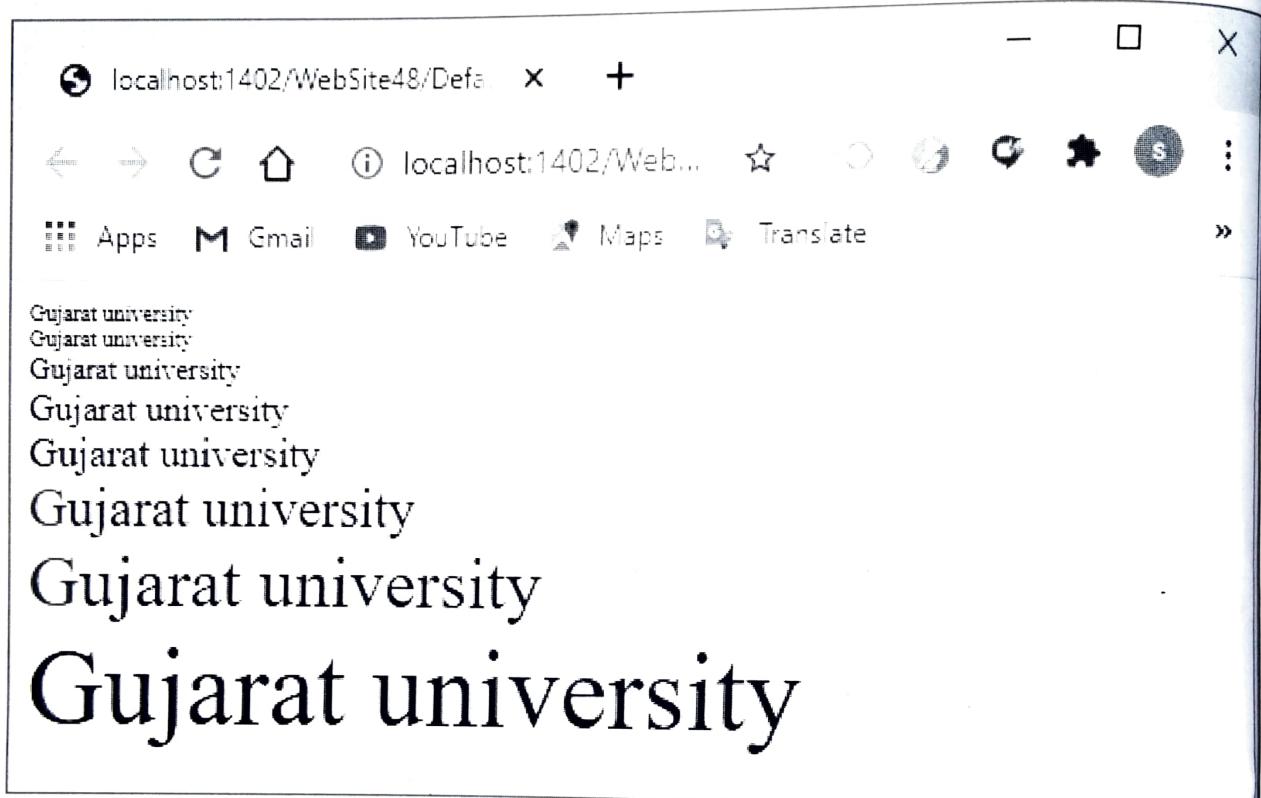
```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <% int i;
                for(i=0;i<8;i++) %>

```

```
{ %>
<font size='<%=i %>'>Gujarat university</font>
<br />
<%} %>
</div>
</form>
</body>
</html>
```



Program : 10

Create web page which will ask the employee personal data education detail, work experience detail with use of different panel for each part. Allow user to click on submit button and display message “Data is successfully submitted” in a new label by adding it at runtime in a panel.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choose option “Add New Item”, select “web Form” option and click on “Add” button.
- Now place 4 panels on the webpage from the ToolBox, one is for Personal Details, one for Educational details, one for Experience Details and one for showing message.

each panel we will take a separate table to get the details from the user. At the end place a button the webpage and change its Text property to "Submit" from the property windows.

The screenshot shows a web browser window with the URL `localhost:1903/WebSite50/Default.aspx`. The page contains a form with the following sections:

- Personal Details:**
 - First Name:
 - Last Name:
 - Email:
 - Mobile No:
- Educational Details:**

	University/Board	Passing Year	Percentage Obtained
S.S.C	<input type="text"/>	<input type="text"/>	<input type="text"/>
H.S.C	<input type="text"/>	<input type="text"/>	<input type="text"/>
Graduate	<input type="text"/>	<input type="text"/>	<input type="text"/>
Post Graduate	<input type="text"/>	<input type="text"/>	<input type="text"/>
- Experience Details:**
 - Company Name:
 - From Date:
 - To Date:
 - Post:

- Write the following code to design form in the source mode.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
```

```
.style1
{
    width: 100%;
}

.style2
{
    width: 142px;
}

.style3
{
    width: 213px;
}

.style4
{
    width: 151px;
}

.style5
{
    width: 171px;
}

.style6
{
    width: 116px;
}

</style>
</head>
<body>

<form id="form1" runat="server">
<div>

<asp:Panel ID="Panel1" runat="server">
    <strong>Personal Details:<br /> </strong>
    <table class="style1">
        <tr>
```

```
<td class="style2">
    First Name:</td>
<td>
    <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
    <td class="style2">
        Last Name:</td>
    <td>
        <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
    <td class="style2">
        Email:</td>
    <td>
        <asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
    <td class="style2">
        Mobile No:</td>
    <td>
        <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
</td>
</tr>
</table>
</asp:Panel>
<br />
<asp:Panel ID="Panel12" runat="server">
```

CC-301 Web Application Development- I (Using C#)

```
<strong>Educational Details:<br /> </strong>
<table class="style1">
    <tr>
        <td class="style6">
            &nbsp;</td>
        <td class="style4">
            University/Board</td>
        <td class="style5">
            Passing Year</td>
        <td>
            Percentage Obtained</td>
    </tr>
    <tr>
        <td class="style6">
            S.S.C</td>
        <td class="style4">
            <asp:TextBox ID="TextBox5"
runat="server"></asp:TextBox>
        </td>
        <td class="style5">
            <asp:TextBox ID="TextBox6"
runat="server"></asp:TextBox>
        </td>
        <td>
            <asp:TextBox ID="TextBox7"
runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="style6">
            H.S.C</td>
        <td class="style4">
            <asp:TextBox ID="TextBox8"
runat="server"></asp:TextBox>
        </td>
```

```
<td class="style5">
    <asp:TextBox ID="TextBox9"
runat="server"></asp:TextBox>
</td>
<td>
    <asp:TextBox ID="TextBox10"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
    <td class="style6">
        Graduate</td>
    <td class="style4">
        <asp:TextBox ID="TextBox11"
runat="server"></asp:TextBox>
</td>
    <td class="style5">
        <asp:TextBox ID="TextBox12"
runat="server"></asp:TextBox>
</td>
    <td>
        <asp:TextBox ID="TextBox13"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
    <td class="style6">
        Post Graduate</td>
    <td class="style4">
        <asp:TextBox ID="TextBox14"
runat="server"></asp:TextBox>
</td>
    <td class="style5">
        <asp:TextBox ID="TextBox15"
runat="server"></asp:TextBox>
</td>

```

```
<asp:TextBox ID="TextBox16"
runat="server"></asp:TextBox>
</td>
</tr>
</table>
</asp:Panel>
<br />
<asp:Panel ID="Panel3" runat="server">
<strong>Experience Details:<br /> </strong>
<table class="style1">
<tr>
<td class="style3">
Company Name:</td>
<td>
<asp:TextBox ID="TextBox17"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style3">
From Date:</td>
<td>
<asp:TextBox ID="TextBox18"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style3">
To Date:</td>
<td>
<asp:TextBox ID="TextBox19"
runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style3">
Post:</td>
```

```
<td>
    <asp:TextBox ID="TextBox20"
runat="server"></asp:TextBox>
</td>
</tr>
</table>
</asp:Panel>
<br />
<asp:Panel ID="Panel4" runat="server">
</asp:Panel>
<br />
<asp:Button ID="Button1" runat="server"
Text="Submit" />
</div>
</form>
</body>
</html>
```

- Now double click on the submit button. This will open a subroutine called "Button1_Click". Write the following code on the click event of the button to dynamically add the label into Panel4.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label lb = new Label();
```

```

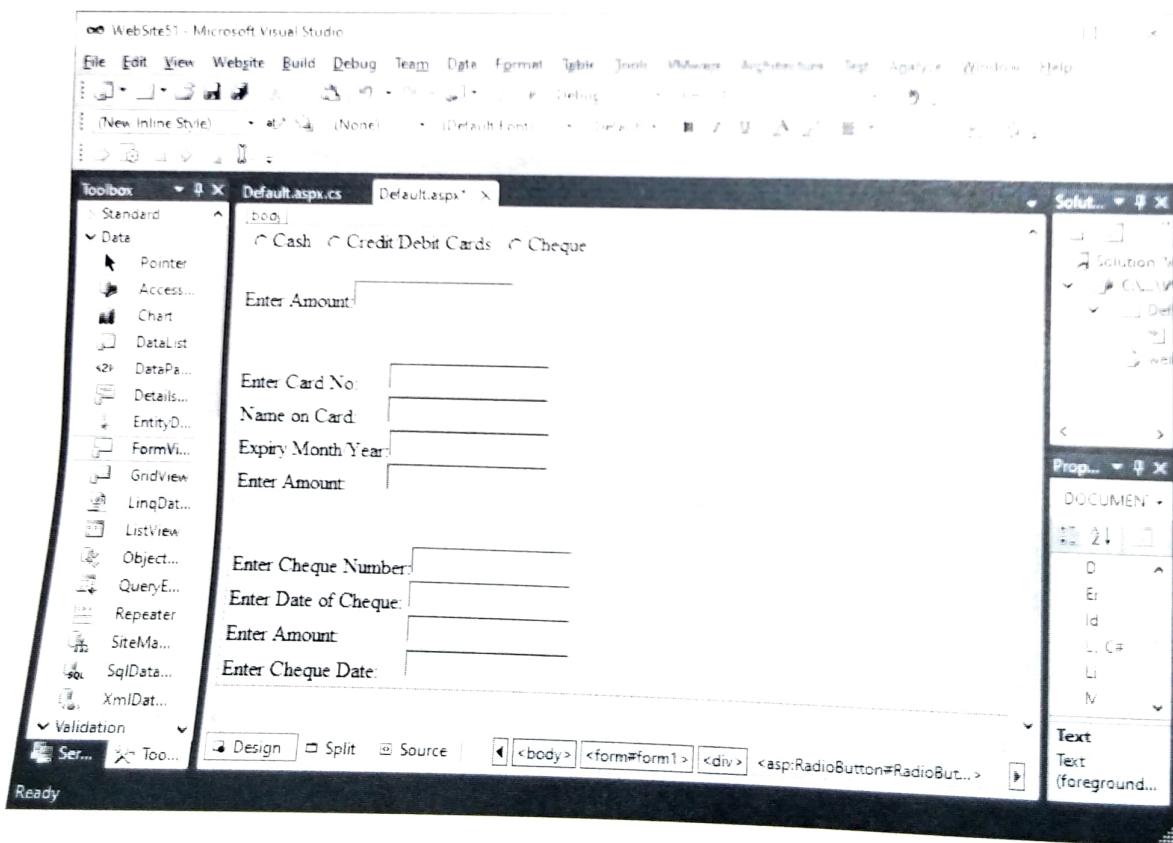
        lb.Text = "Your data is saved Successfully";
        Panel4.Controls.Add(lb);
    }
}

```

Program : 11

Create web page which will ask payment detail of customer purchase, this detail either in Cash or Credit/Debit card or Cheque. According to the payment mode panel control will display and accept payment detail and display all that detail in next page using label control.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choose option “Add New Item”, select “web Form” option and click on “Add” button.
- Now add 3 radio buttons to the webpage and change its Text property to “Cash”, “Credit/Debit Cards”, and “Cheque” from the property window.
- Set the “GroupName” property for all 3 RadioButtons to be “PaymentMode”. So, the user can select any one from these radio buttons.
- Add 3 panels from the ToolBox to a web page, and design it as below.



- Set the Visible property of all panels to be “false”, from the property window. Now double click on the RadioButton1. This is open subroutine “RadioButton1_CheckedChanged” write the following code on the CheckChanged event of the RadioButton1. Similarly, also add the code on the CheckChanged event of RadioButton2 and RadioButton3.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void RadioButton1_CheckedChanged(object sender, EventArgs e)
    {
        if (RadioButton1.Checked)
        {
            Panel1.Visible = true;
            Panel2.Visible = false;
            Panel3.Visible = false;
        }
    }

    protected void RadioButton2_CheckedChanged(object sender, EventArgs e)
    {
        if (RadioButton2.Checked)
        {
            Panel1.Visible = false;
            Panel2.Visible = true;
        }
    }
}
```

```

        Panel3.Visible = false;
    }

}

protected void RadioButton3_CheckedChanged(object
sender, EventArgs e)
{
    if (RadioButton3.Checked)
    {
        Panel1.Visible = false;
        Panel2.Visible = false;
        Panel3.Visible = true;
    }
}

```

- Run the web page in the browser and click on each RadioButton and see one by one panel (one at a time) will be opened automatically.

Program : 12

Write a program to set the following properties of Label control using internal css class.

- **Background-color as green,**
- **Border style as solid**
- **Border color as blue**
- **Border width as 2px**
- **Text as "Hello!"**

When user moves mouse over the label, its background color should change it to yellow. Add one more web form which contains button. When user clicks on it change its fore color as pink using external css.

- Create a “ASP.NET Empty Website” using visual studio and add a webpage default.aspx by doing Right Click on the website in the Solution Explorer and choosing option “Add New Item”, select “web Form” option and click on “Add” button.
- Now add a style sheet class in the <head> tag. (place the following code between <head> and </head> in the “Source” of the webpage.)

```

.myStyle
{
    background-color:Green;
    border: 2px solid blue;
}
</style>

```

- Add a label control to a web page and set Text property to “Hello” and CSSClass property to “myStyle”.
- The complete source code of the file is as given below.

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

```

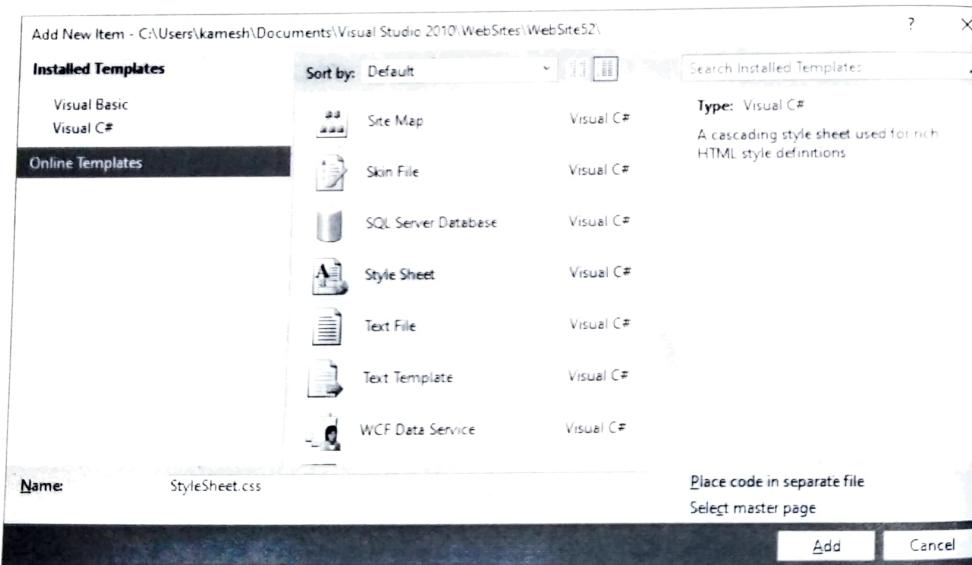
```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Internal Style Sheet Demo</title>
    <style>
        .myStyle
        {
            background-color:Green;
            border: 2px solid blue;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
<asp:Label ID="Label1" runat="server" Text="Hello"
CssClass="myStyle"
onmouseover="this.style.background='yellow';
this.style.border= '2px solid blue' "

```

```
></asp:Label>
</div>
</form>
</body>
</html>
```

- The code written '**onmouseover="this.style.background='yellow'; this.style.borderColor='2px solid blue' "**', will change the background color of the label on mouse move event.
- Now, Right Click on the web page in the Solution Explorer and choose option "Add New Item". From the "Add New Item" dialog box select option "Style Sheet" and then click on Add button. This action will add one file called "StyleSheet.css" to your website.



- Add the following code to your "StyleSheet.css" file.

```
body {
}
.myStyle
{
    background-color: #FF3399;
}
```

- Instead of writing the code you can also use "Build Style" option, which you can get on the top-left side, in the tool bar.
- Now add another web page called "default2.aspx" to your website. Open the page in to the design mode. Now drag and drop your StyleSheet.css file to your web page. This will add one line of code in the "Source" of your webpage, in the <Head> section.

```
<link href="StyleSheet.css" rel="stylesheet" type="text/css" />
```

- Place a button on the webpage and write the following code on the click event of the button.

```
Button1.CssClass = "myStyle";
```

- Run default2.aspx and click on the button. You will observe that the background color of the button gets change to 'pink' color.

- The source code of the default2.aspx is as given below

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
        </div>
    </form>
</body>
```

</html>

- The complete code of the default.aspx.cs file is given below:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Button1.CssClass = "myStyle";
    }
}
```