BCS 2203 | WEB APPLICATION DEVELOPMENT

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

ASP.NET

C# Programming

DR ZURIANI
MUSTAFFA

AL - FATIHAH

CH 2| ASP.NET PROGRAMMING MODEL

COURSE OUTCOMES

- Using Visual Studio/Visual Web Developer to create new ASP.NET websites and web pages
- Viewing an ASP.NET page through a web browser
- Using conditional statements
- Using functions

ASP.NET PROGRAMMING MODEL

- ASP.NET pages are composed of two portions:
  - An HTML portion
  - A source code portion.

- The HTML portion can be composed of both static HTML and Web controls.

- Web controls are programmatically accessible chunks of HTML that enable developers to modify the HTML sent to the browser using C# or VB.NET code.

- Web controls also provide a means by which user input can be collected.

- Web controls serve as the bridge between the HTML and source code portion.

CREATING THE HTML PORTION OF AN ASP.NET PAGE

- Visual Web Developer offers three views of an ASP.NET page's HTML portion.

- HTML content and Web controls may be added to an ASP.NET page through any of these views:

1. Source View

You can either type in the HTML and Web control syntax by hand or drag the desired HTML elements and Web controls from the Toolbox onto the Source View

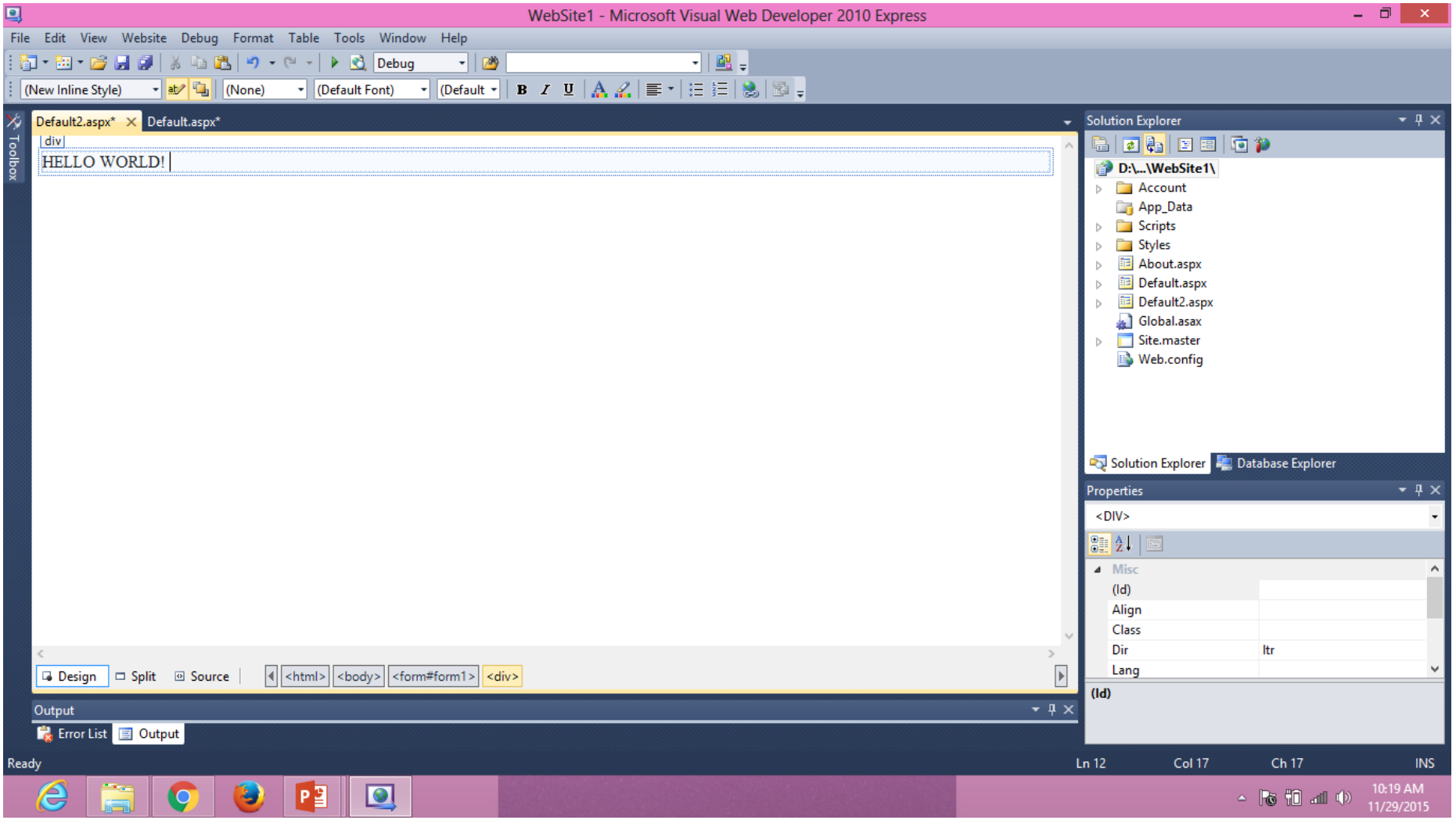
2. Design View

Present a What You See Is What You Get (WYSIWYG). You can add text to the page by typing, but HTML elements and Web controls can be added only by dragging them from the Toolbox and dropping them onto the design surface.

3. Split View

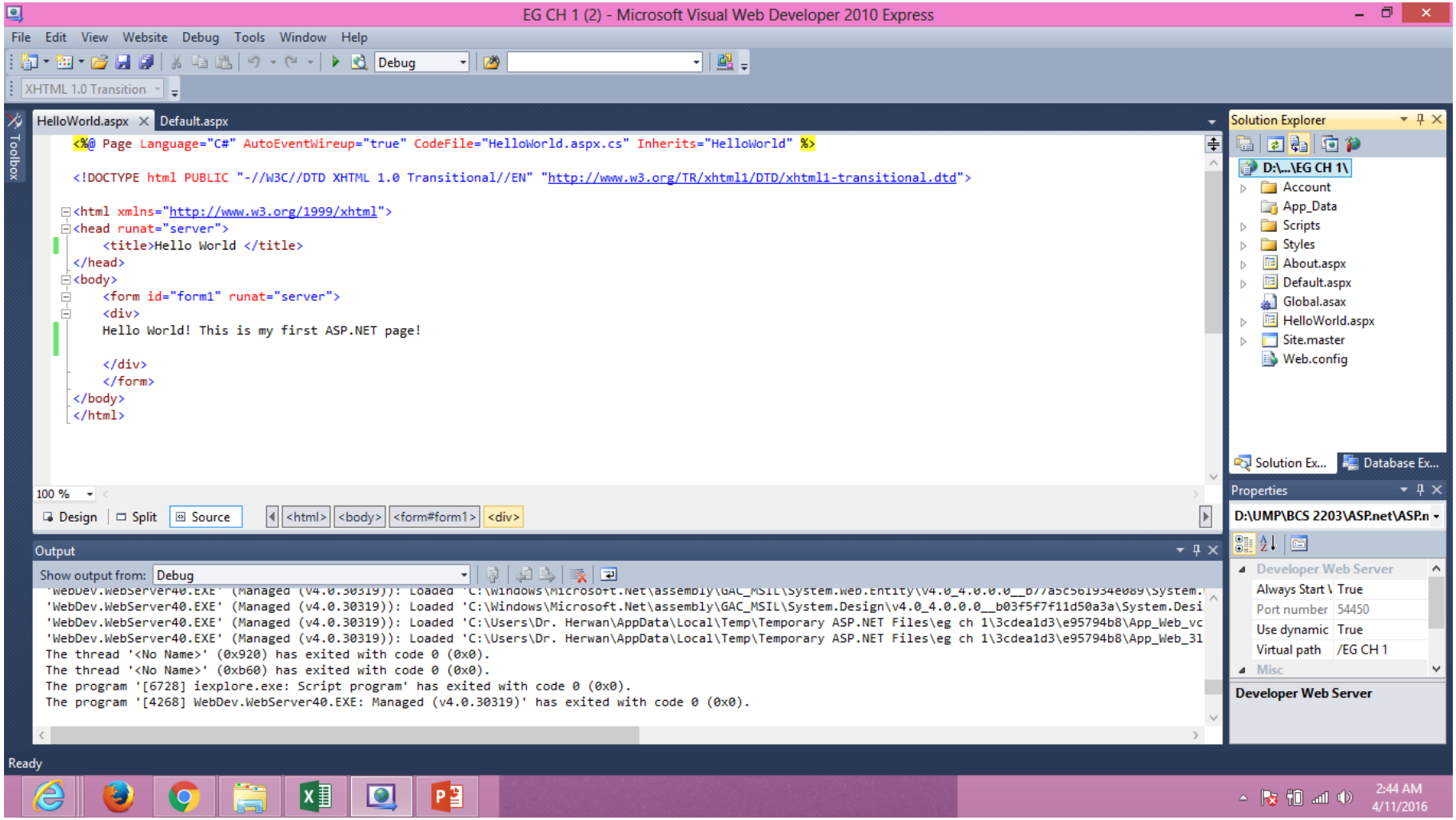
Shows both Source and Design views.

DESIGN VIEW


WebSite1 - Microsoft Visual Web Developer 2010 Express
File Edit View Website Debug Format Table Tools Window Help
Debug
(New Inline Style)
(None)
(Default Font)
(Default
Default2.aspx*
Default.aspx*
Toolbox
div
HELLO WORLD!
Design
Split
Source
<html>
<body>
<form#form1>

Solution Explorer
D:\...\WebSite1\
Account
App_Data
Scripts
Styles
About.aspx
Default.aspx
Default2.aspx
Global.asax
Site.master
Web.config
Solution Explorer
Database Explorer
Properties
<DIV>
Misc
(Id)
Align
Class
Dir
ltr
Lang
(Id)
Output
Error List
Output
Ready
Ln 12
Col 17
Ch 17
INS
10:19 AM
11/29/2015

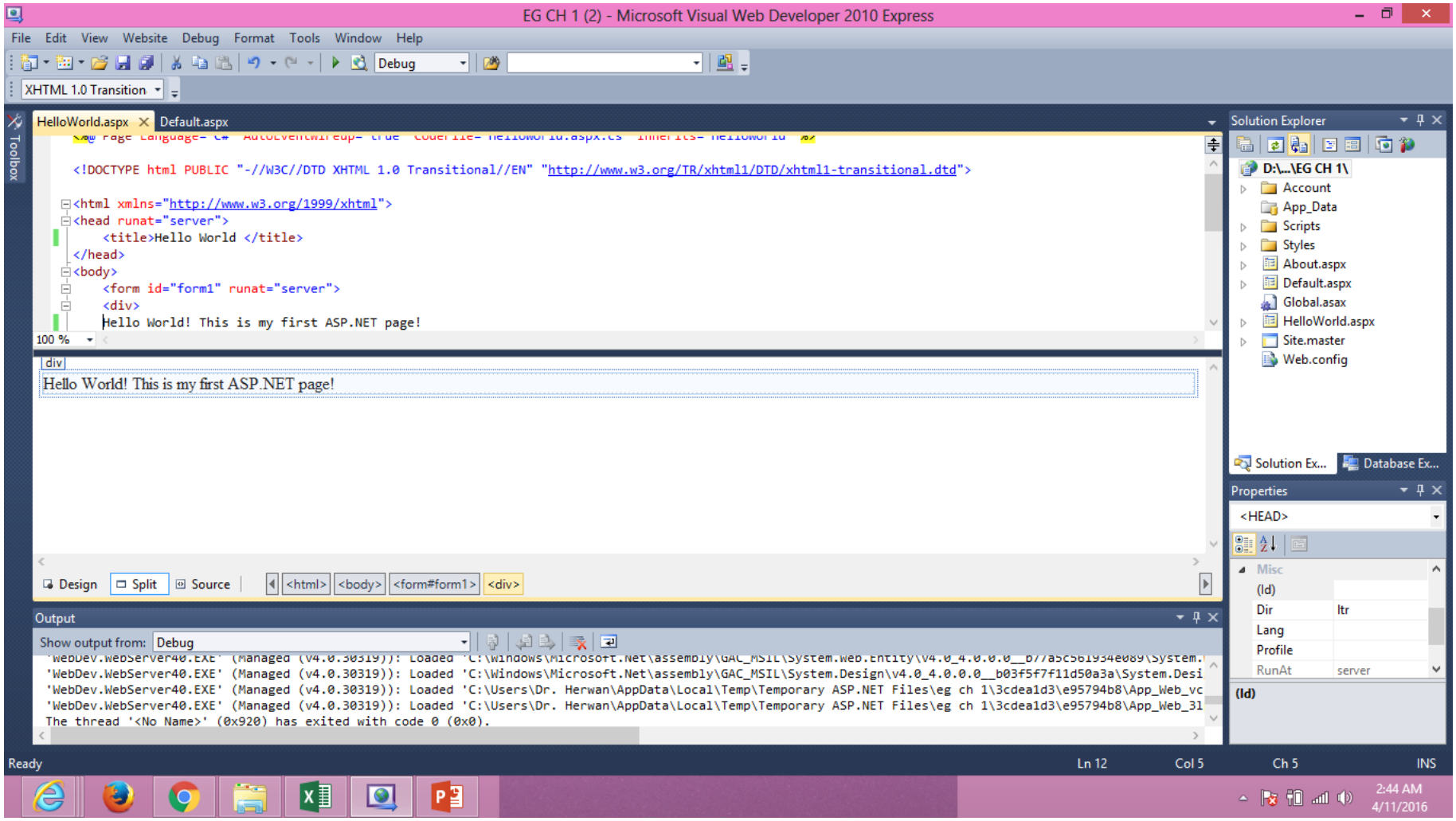
SOURCE VIEW


EG CH 1 (2) - Microsoft Visual Web Developer 2010 Express
File Edit View Website Debug Tools Window Help
Debug
XHTML 1.0 Transition
HelloWorld.aspx
Default.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="HelloWorld.aspx.cs" Inherits="HelloWorld" %>
<!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>Hello World </title>
</head>
<body>
<form id="form1" runat="server">

Hello World! This is my first ASP.NET page!

</form>
</body>
</html>
100 %
Design Split Source
<html> <body> <form#form1>
Solution Explorer
D:\...\EG CH 1\
Account
App_Data
Scripts
Styles
About.aspx
Default.aspx
Global.asax
HelloWorld.aspx
Site.master
Web.config
Solution Ex... Database Ex...
Properties
D:\UMP\BCS 2203\ASP.net\ASP.n
Developer Web Server
Always Start \ True
Port number 54450
Use dynamic True
Virtual path /EG CH 1
Misc
Developer Web Server
Output
Show output from: Debug
'WebDev.WebServer40.EXE' (Managed (v4.0.30319)): Loaded 'C:\Users\Dr. Herwan\AppData\Local\Temp\Temporary ASP.NET Files\eg ch 1\3cdea1d3\e95794b8\App_Web_vc
'WebDev.WebServer40.EXE' (Managed (v4.0.30319)): Loaded 'C:\Users\Dr. Herwan\AppData\Local\Temp\Temporary ASP.NET Files\eg ch 1\3cdea1d3\e95794b8\App_Web_31
The thread '<No Name>' (0x920) has exited with code 0 (0x0).
The thread '<No Name>' (0xb60) has exited with code 0 (0x0).
The program '[6728] iexplore.exe: Script program' has exited with code 0 (0x0).
The program '[4268] WebDev.WebServer40.EXE: Managed (v4.0.30319)' has exited with code 0 (0x0).
Ready
2:44 AM
4/11/2016

SPLIT VIEW


EG CH 1 (2) - Microsoft Visual Web Developer 2010 Express
File Edit View Website Debug Format Tools Window Help
Debug
XHTML 1.0 Transition
HelloWorld.aspx
Default.aspx
<!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>Hello World </title>
</head>
<body>
<form id="form1" runat="server">

Hello World! This is my first ASP.NET page!
100 %
div
Hello World! This is my first ASP.NET page!
Design Split Source
<html> <body> <form#form1>
Solution Explorer
D:\...\EG CH 1\
Account
App_Data
Scripts
Styles
About.aspx
Default.aspx
Global.asax
HelloWorld.aspx
Site.master
Web.config
Solution Ex...
Database Ex...
Properties
<HEAD>
Misc
(Id)
Dir ltr
Lang
Profile
RunAt server
(Id)
Output
Show output from: Debug
'WebDev.WebServer40.EXE' (Managed (v4.0.30319)): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Design\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Desi
'WebDev.WebServer40.EXE' (Managed (v4.0.30319)): Loaded 'C:\Users\Dr. Herwan\AppData\Local\Temp\Temporary ASP.NET Files\eg ch 1\3cdea1d3\e95794b8\App_Web_vc
'WebDev.WebServer40.EXE' (Managed (v4.0.30319)): Loaded 'C:\Users\Dr. Herwan\AppData\Local\Temp\Temporary ASP.NET Files\eg ch 1\3cdea1d3\e95794b8\App_Web_31
The thread '<No Name>' (0x920) has exited with code 0 (0x0).
Ready
Ln 12 Col 5 Ch 5 INS
2:44 AM
4/11/2016

EXAMPLE

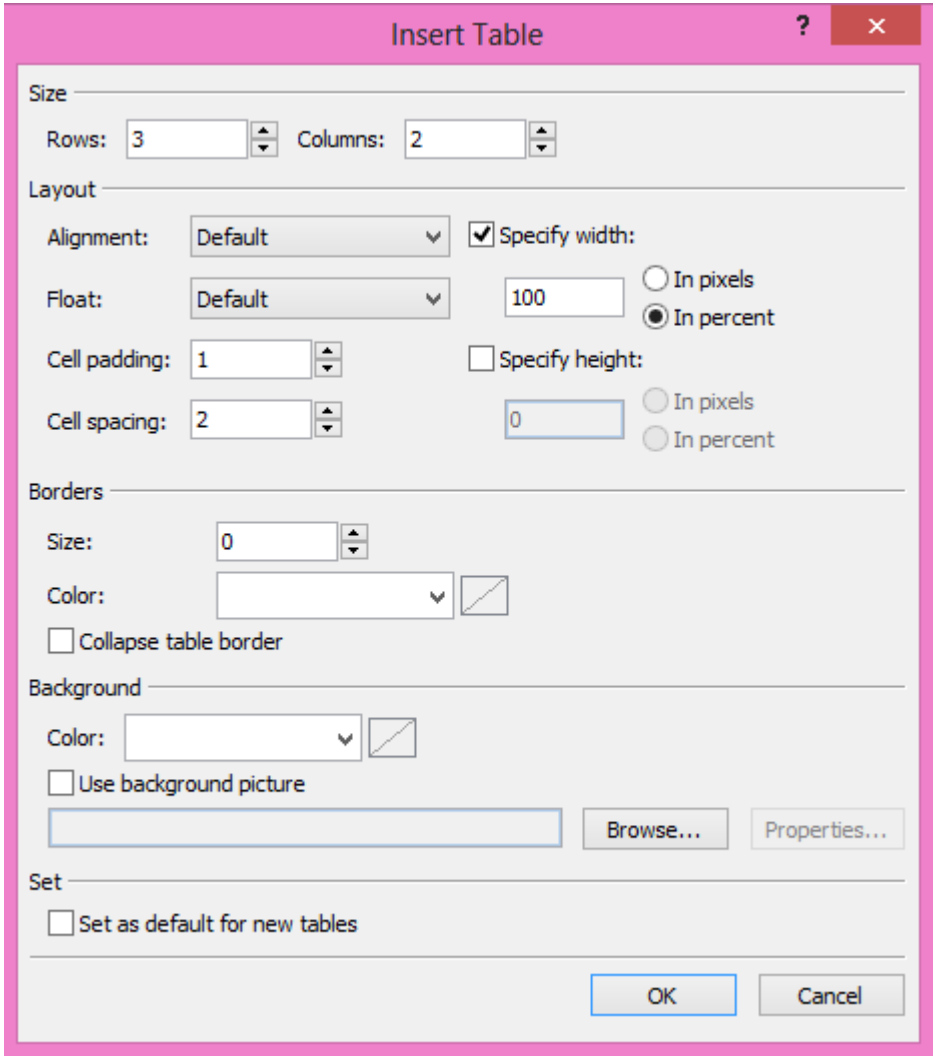
To illustrate adding content to the HTML portion of an ASP.NET page, let's create a simple ASP.NET website with a single web page.

1. Add an HTML <table> to the page

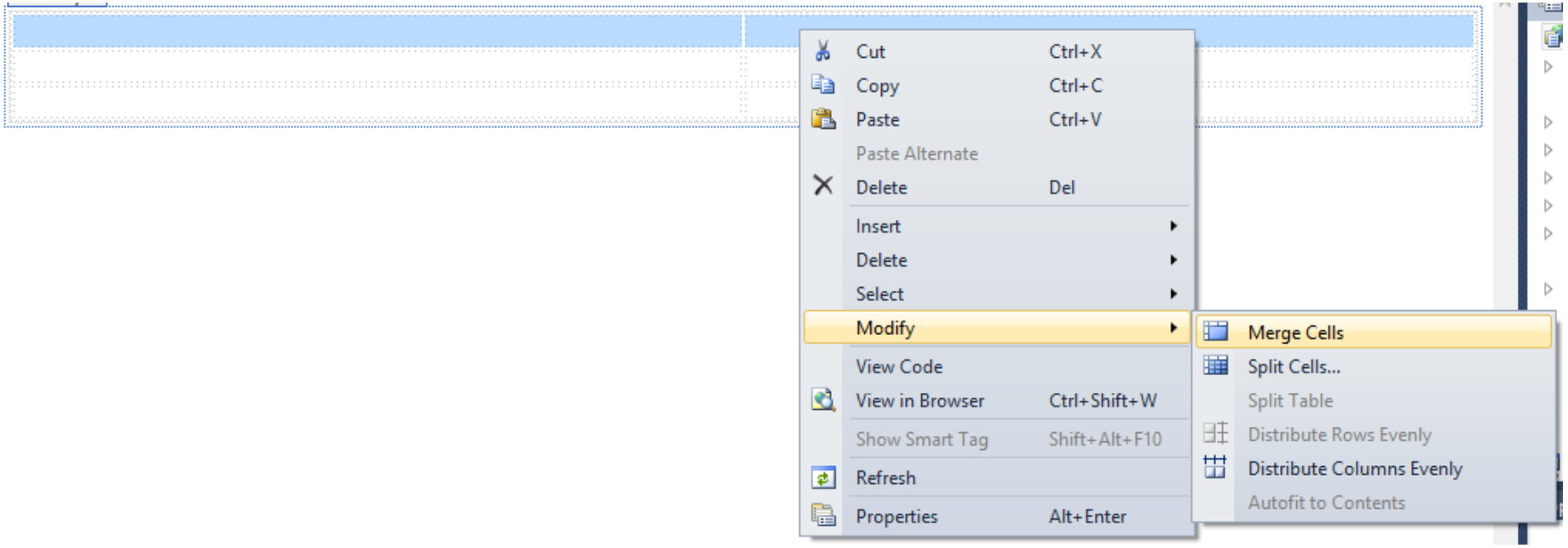
2. Because we won't need the element, go ahead and remove it (delete the opening and closing tags from the Source View, by right-clicking in the region in the Design View and choosing the Delete menu option or by selecting the region in the Design view and hitting the Delete key.

- Set the focus in the Design View so that the cursor is within the <form> element.

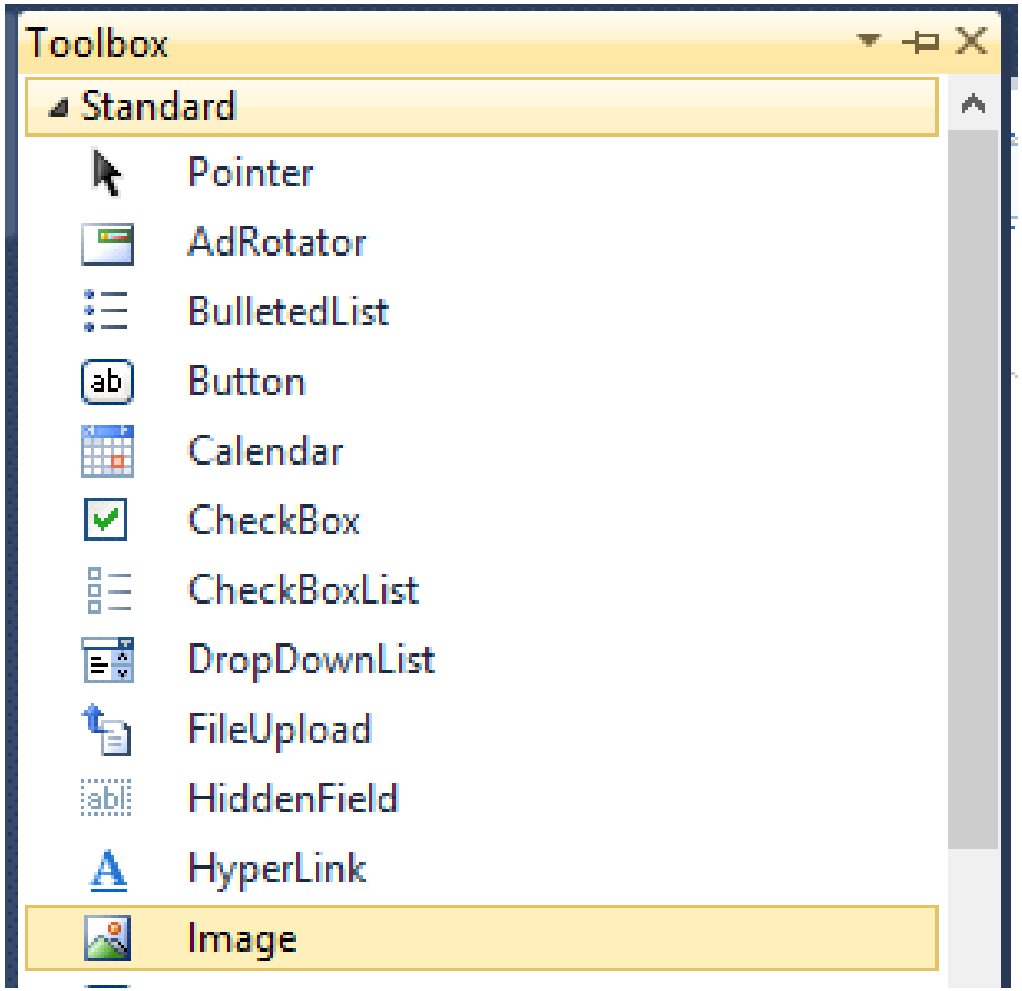
- Go to the Table menu -> Insert Table

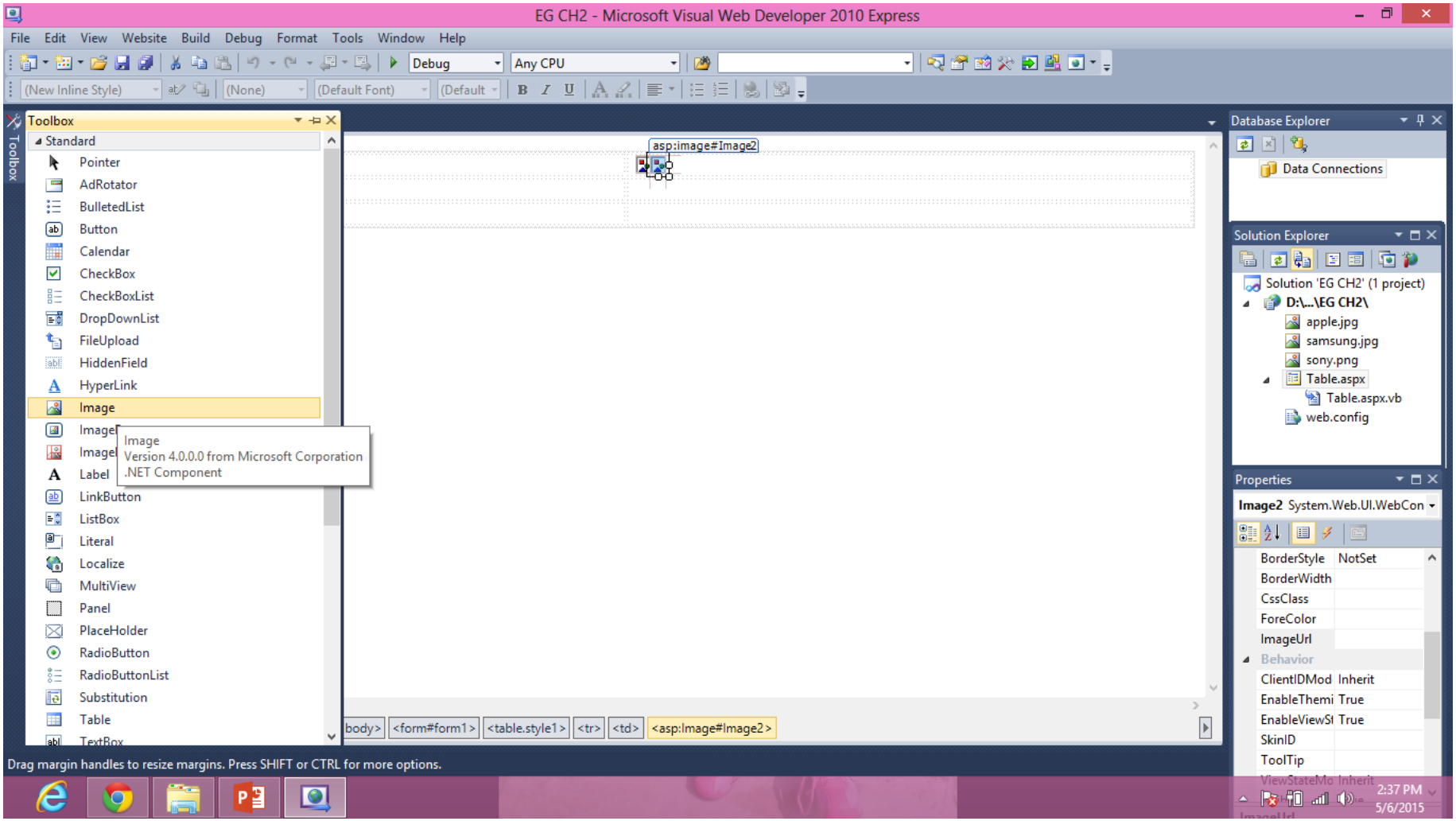


Merge the first row by selecting Modify – Merge Cells

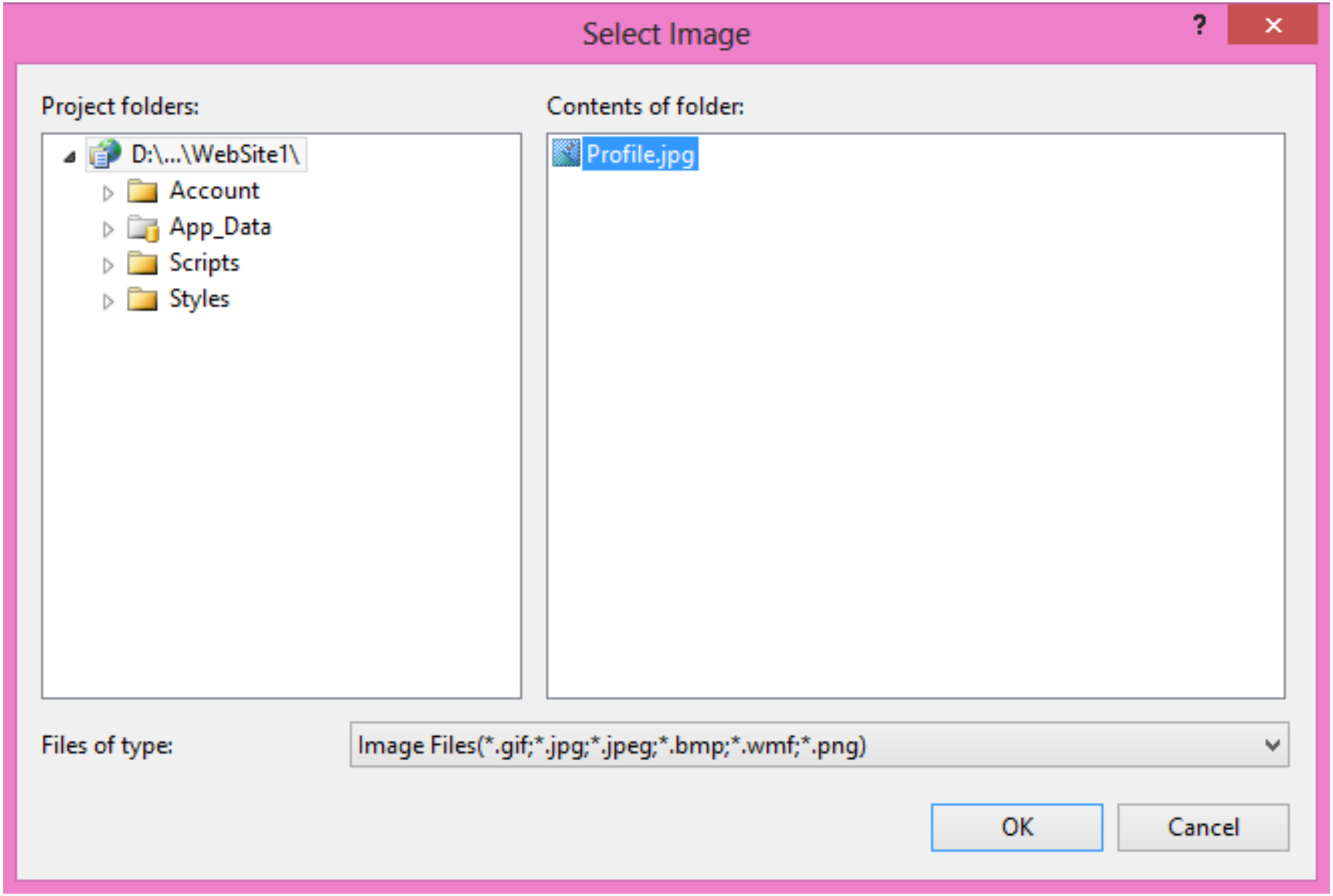


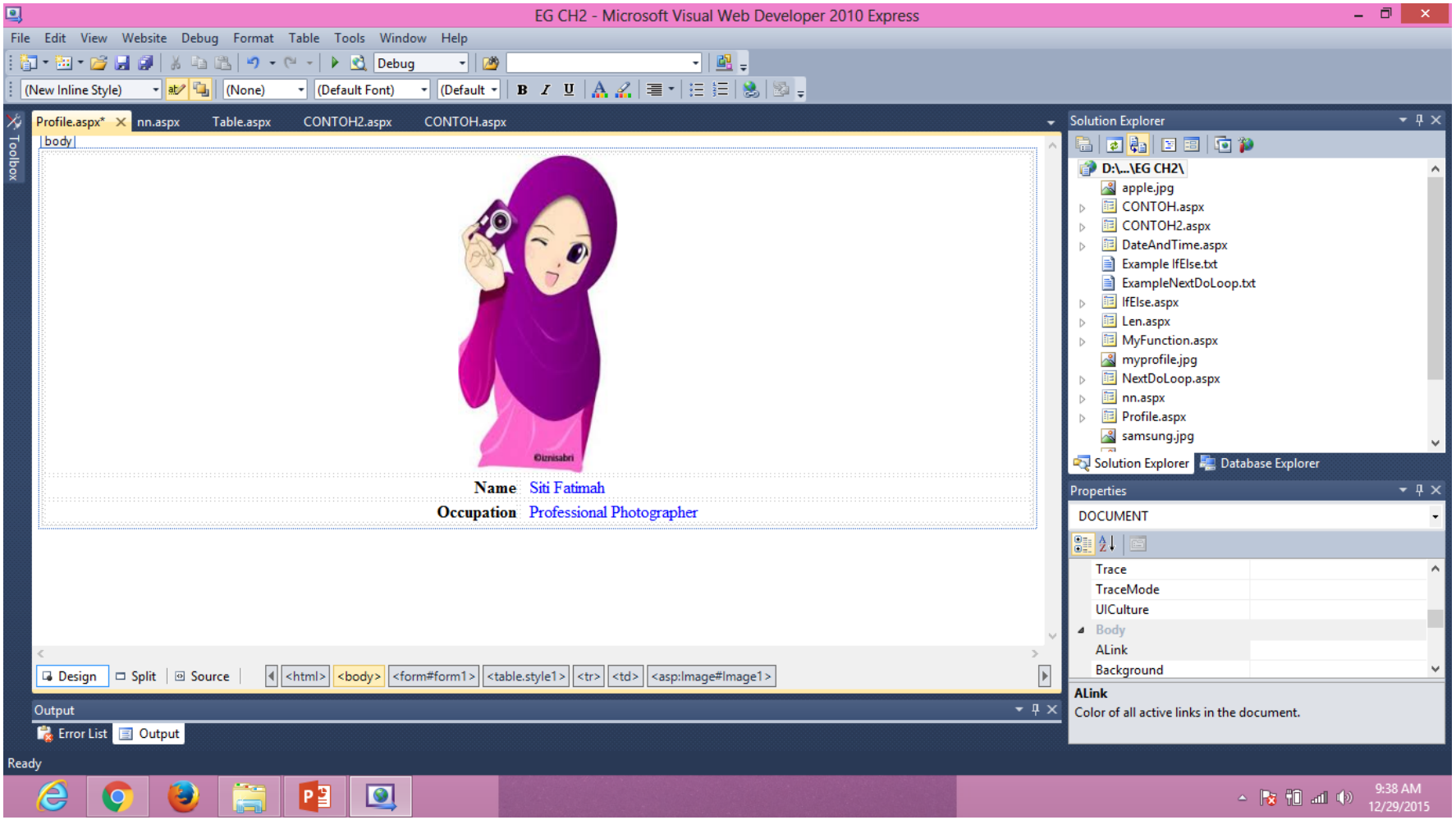
For first row:
Toolbox - Image



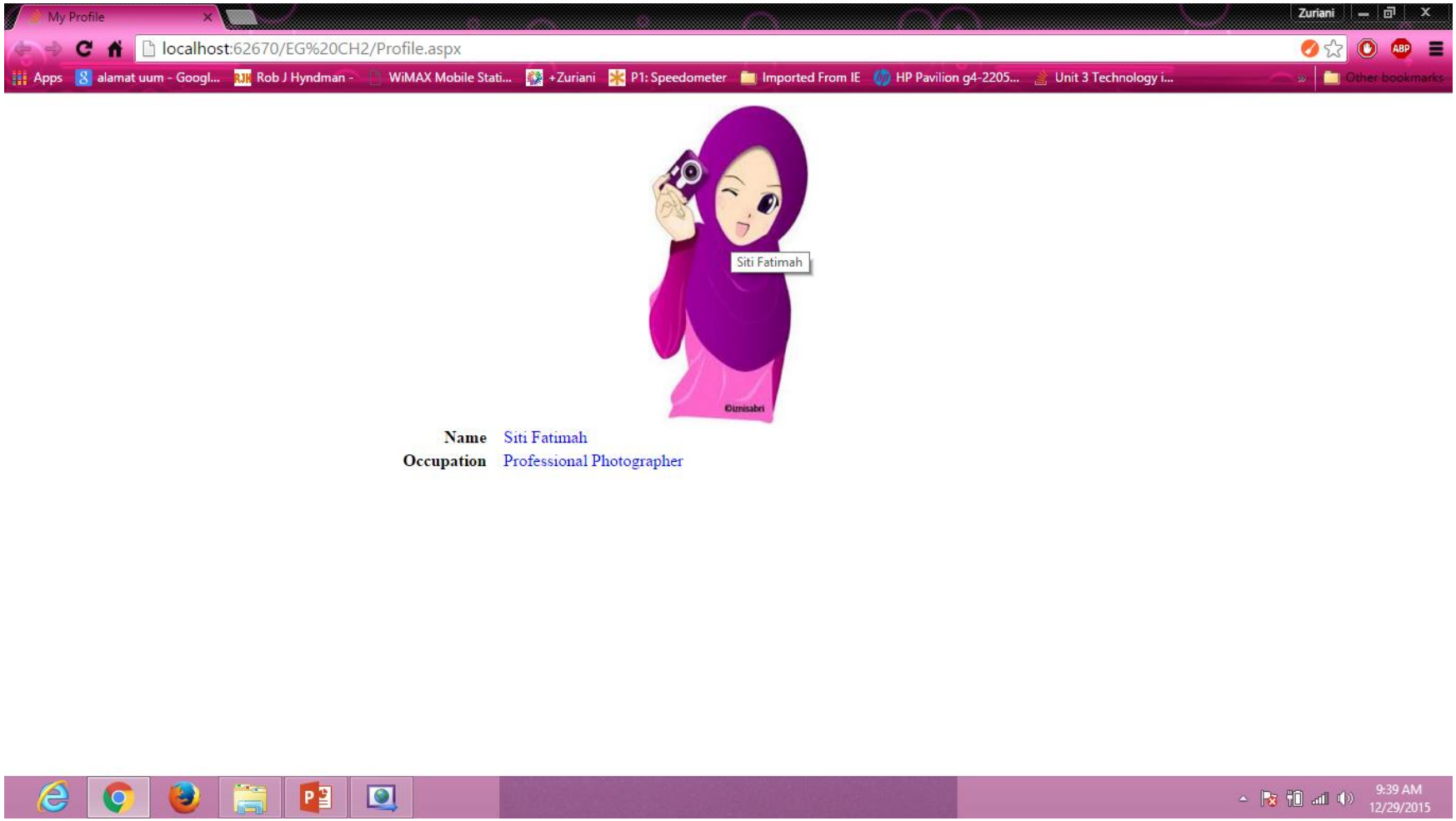


- After you download the images and saved them to your website's folder, edit the ImageUrl properties


Select Image
Project folders:
D:\...\WebSite1\
Account
App_Data
Scripts
Styles
Contents of folder:
Profile.jpg
Files of type:
Image Files(*.gif;*.jpg;*.jpeg;*.bmp;*.wmf;*.png)
OK
Cancel



- To adjust the image's height and width, you can alter its Height and Width properties, eg: Height: 300 pixels, Width: 150px



Now we've created the HTML content which includes both static HTML, such as <table> and Image Web Controls

SOURCE CODE PORTION OF AN ASP.NET PAGE

- Source code portion can be put In a separate file or In a <script> block in the same file

  - In a separate file
    The ASP.NET page is composed of two separate files, PageName.aspx (contains the HTML portion) and PageName.aspx.cs (contains the source code).

  - In a <script> block in the same file
    The HTML and source code portions can reside in the same file. In this scenario, the source code is placed within a <script runat="server"> element

SOURCE CODE PORTION OF AN ASP.NET PAGE

- Advantage of separate file approach

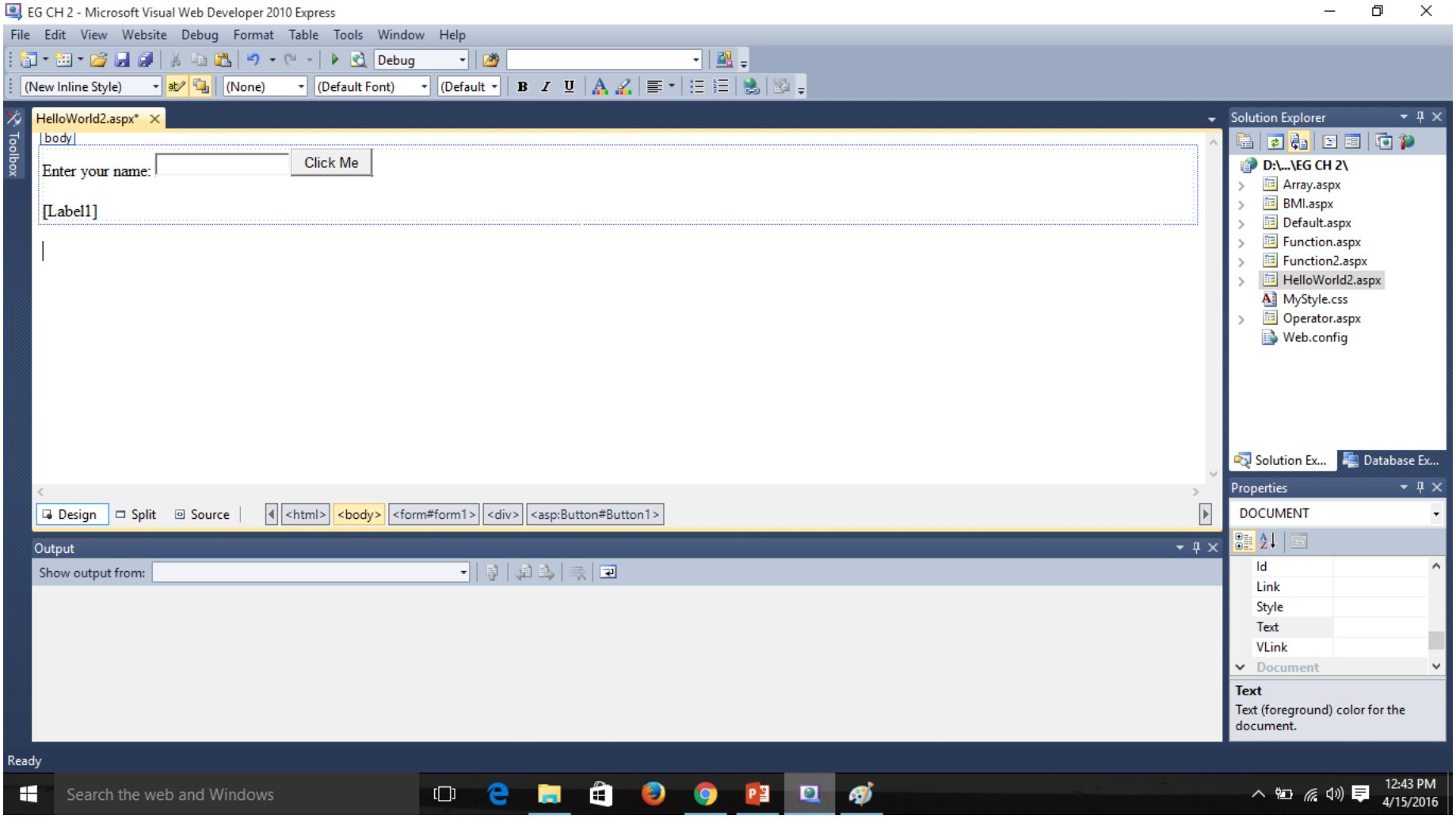
  - Provides a cleaner separation between in HTML and source code portions.

```
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)
    {

    }
}
```

These lines are the bare minimum code that must exist in an ASP.NET page's source code portion.

Simple example

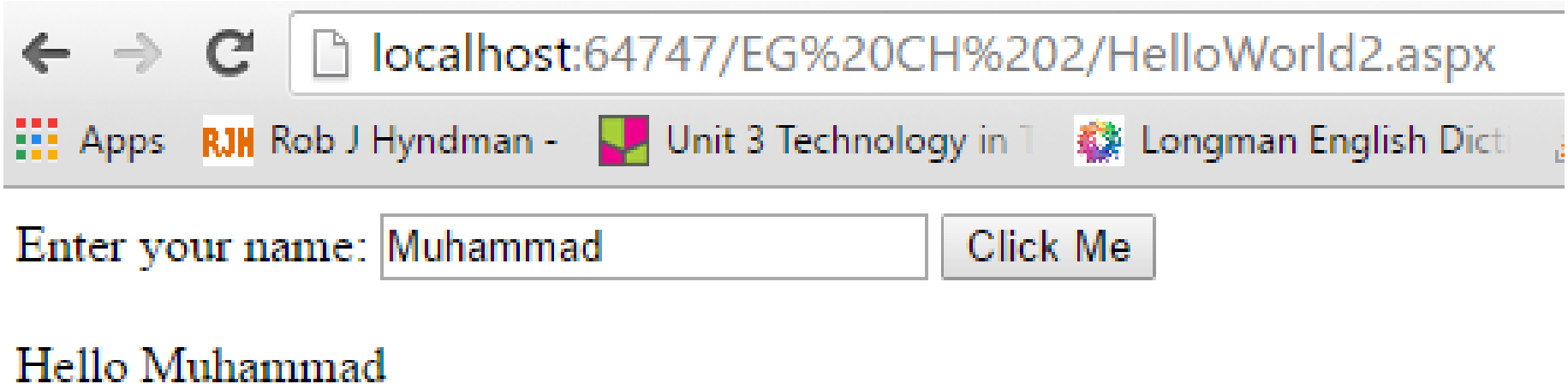
- 1. In the design view, drag and drop a Button web control and Label web control
- Set the Text Property of the Button web control to “Click Me”.
- For the Label web control, clear it’s Text Property.
- Feel free to rename the ID for both web controls.


EG CH 2 - Microsoft Visual Web Developer 2010 Express
File Edit View Website Debug Format Table Tools Window Help
Debug
(New Inline Style)
(None)
(Default Font)
(Default
HelloWorld2.aspx*
body
Enter your name:
Click Me
[Label1]
Design
Split
Source
<html>
<body>
<form#form1>

<asp:Button#Button1>
Output
Show output from:
Solution Explorer
D:\...\EG CH 2\
Array.aspx
BMI.aspx
Default.aspx
Function.aspx
Function2.aspx
HelloWorld2.aspx
MyStyle.css
Operator.aspx
Web.config
Solution Ex...
Database Ex...
Properties
DOCUMENT
Id
Link
Style
Text
VLink
Document
Text
Text (foreground) color for the document.
Ready
Search the web and Windows
12:43 PM
4/15/2016

In the Button_Click event handler, enter the following codes:

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = "Hello " + TextBox1.Text;
}
```



HelloWorld2.aspx

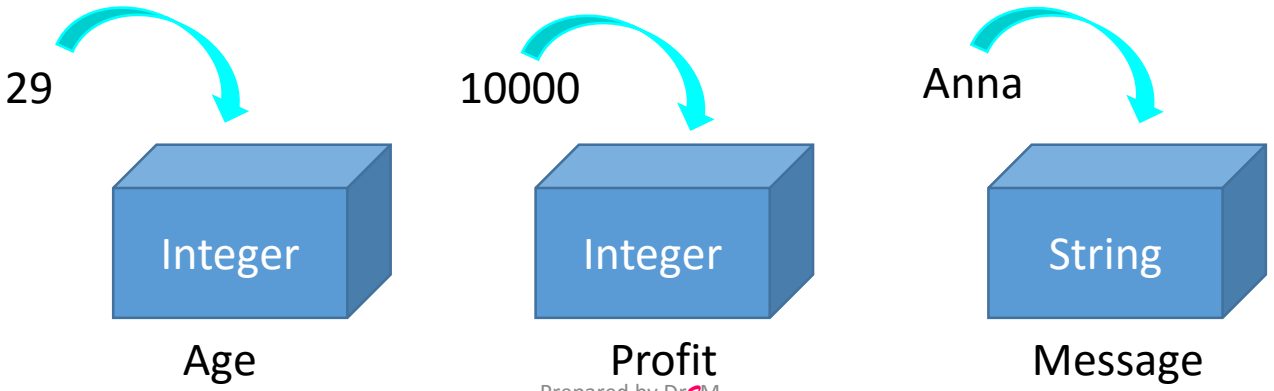
Concepts common to all programming languages

- A means to store data temporarily
- A set of operators that can be applied to the data stored in variables
  - ✓One such operator is +, which sums the values of two variables

DECLARING AND USING VARIABLES

- A variable is a location in the computer's memory where you can temporarily store information, such as number or a string

- 3 components of variables:
  - A value, eg: 5, "Hello World" etc
  - A name which is used to refer to the value of the variable
  - A type, which indicates what types of values can be stored. Eg: a variable of type Integer can store values such as 5, -858. A variable of type String can store values such as "Anna" etc



DEFINING VARIABLES

```
<data_type> <variable_list>;
```

Here, data_type must be a valid C# data type including char, int, float, double, or any user-defined data type, and variable_list may consist of one or more identifier names separated by commas.

Some valid variable definitions are shown here:

```
int i, j, k;
char c, ch;
float f, salary;
double d;
```

INITIALIZING VARIABLES

You can initialize a variable at the time of definition as

```
int i =100;
```

Some examples are:

```
int d = 3, f = 5; /* initializing d and f. */
byte z = 22; /* initializes z. */
double pi = 3.14159; /* declares an approximation of pi. */
char x = 'x'; /* the variable x has the value 'x'. */
```

DEFINING CONSTANTS

```
const <data_type> <constant_name> = value
```

Example:

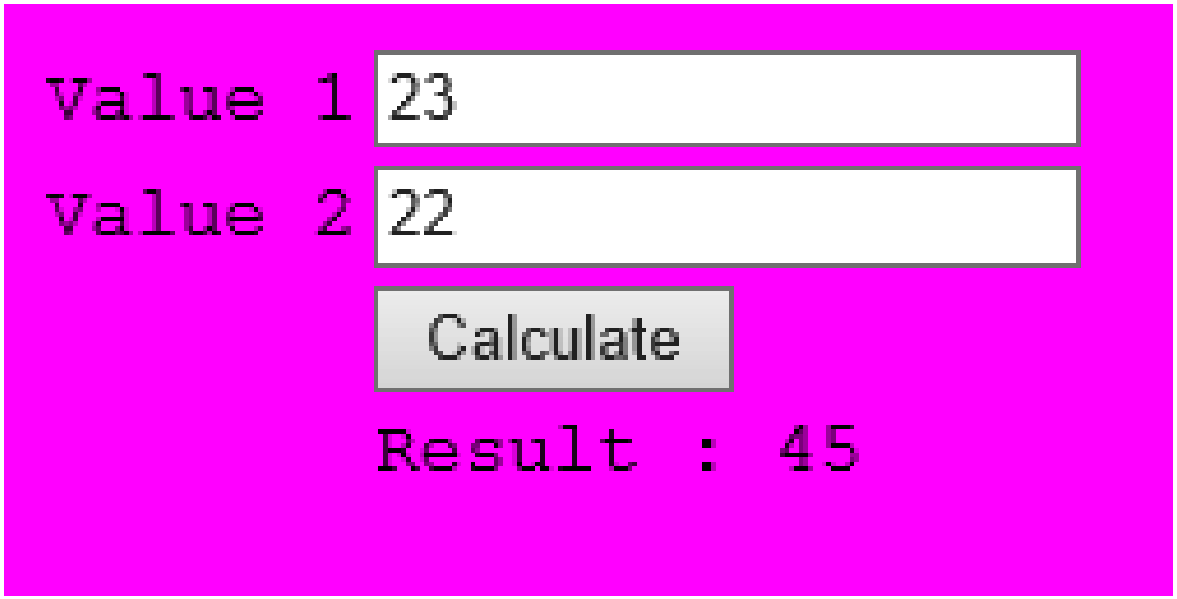
```
const Double pi = 3.142;

Label1.Text = "Pie is equal to : " + pi;
```

ARITHMETIC OPERATORS

Operator	Description	Example
+	Adds two operands	A + B = 30
-	Subtracts second operand from the first	A - B = -10
*	Multiplies both operands	A * B = 200
/	Divides numerator by de-numerator	B / A = 2
%	Modulus Operator and remainder of after an integer division	B % A = 0
++	Increment operator increases integer value by one	A++ = 11
--	Decrement operator decreases integer value by one	A-- = 9

ARITHMETIC OPERATOR: EXAMPLE

```
int Value1 = Convert.ToInt16(txtValue1.Text);
int Value2 = Convert.ToInt16(txtValue2.Text);

int Result = Value1 + Value2;
lblResult.Text = Result.ToString();
```



Operator.aspx

EXERCISE

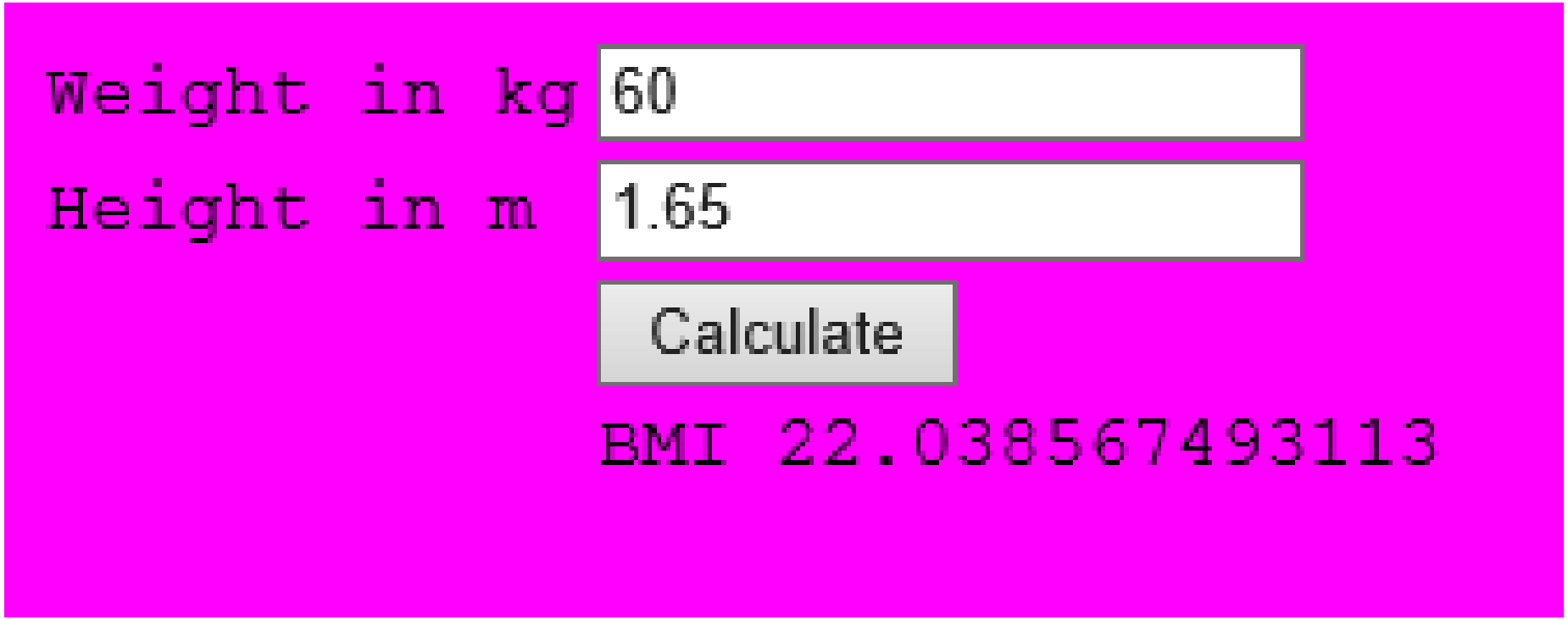
Based on the given formula, calculate your BMI.

$$BMI = \frac{\text{Weight in Kilograms}}{(\text{Height in Meters}) \times (\text{Height in Meters})}$$

*** Convert Weight & Height to Double

*** Convert result (BMI) to String

EXAMPLE OF THE OUTPUT



BMI.aspx

RELATIONAL OPERATORS

Following table shows all the logical operators supported by C#. Assume variable **A** holds Boolean value true and variable **B** holds Boolean value false, then:

Operator	Description	Example
==	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

LOGICAL OPERATORS

Following table shows all the logical operators supported by C#. Assume variable **A** holds Boolean value true and variable **B** holds Boolean value false, then:

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non zero then condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non zero then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true.

CONTROL STRUCTURES

Control structures	Description
Conditional control structure	Executes a set of instructions only if some condition is met
Looping control structures	Repeatedly execute a set of instructions until some condition is met
Modularizing control structures	Group sets of instruction into modules that can be invoked at various places in the program

IF ELSE STATEMENT

The conditional statement IF-ELSE is using for check the conditions that we provided in the head of if statement and making decision based on that condition. The conditional statement examining the data using comparison operators as well as logical operators.

Syntax:

```
if (condition)
        statement;
```

IF ELSE STATEMENT

- If the given condition is true then the control goes to the body of if block , that is the program will execute the code within if block. If the condition is false then the control goes to next level, that is if you provide else block the program will execute the code block of else statement, otherwise the control goes to next line of code.

- The else statement is optional , so you can use if statement without else statement.

Syntax:

```
if (condition)
        statement;
else
        statement;
```

IF ELSE STATEMENT

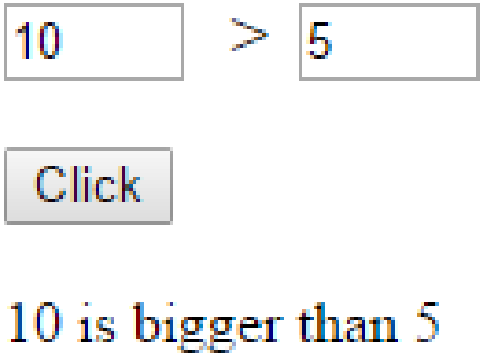
If you want to check more than one conditions then you can use else-if statement

Syntax:

```
if (condition)
        statement;
else if (condition)
        statement;
else
        statement;
```

IF ELSE STATEMENT

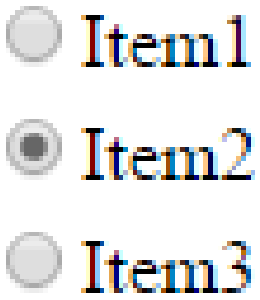
```
protected void Button1_Click(object sender, EventArgs e)
{
    int Val1 = Convert.ToInt16(TextBox1.Text);
    int Val2 = Convert.ToInt16(TextBox2.Text);

    if (Val1 > Val2)
    {
        Label1.Text = Val1 + " is bigger than " + Val2;
    }

    else
    {
        Label1.Text = Val2 + " is bigger than " + Val1;
    }
}
```



SWITH STATEMENT

```
protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    string myItems = RadioButtonList1.SelectedItem.Text;

    switch (myItems)
    {
        case ("Item1"):
            Label1.Text = "This item is unique";
            break;
        case ("Item2"):
            Label1.Text = "This item comes in pair";
            break;
        case ("Item3"):
            Label1.Text = "This item is not for sale";
            break;
    }

}
```

SWITH STATEMENT



RadioButtonList
Set the properties
AutoPostBack=True

This item comes in pair

LOOPING CONTROL STRUCTURE

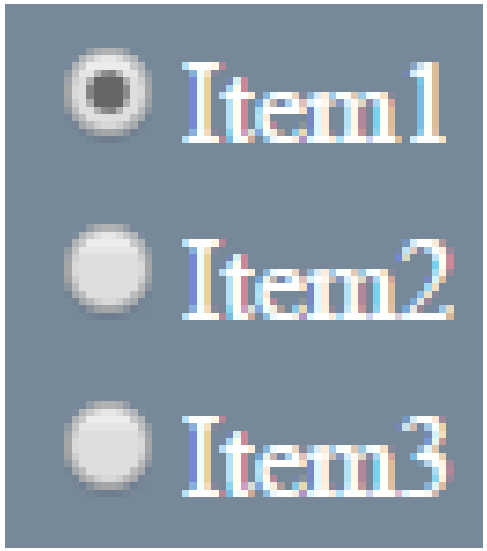
Universiti Malaysia PAHANG
Engineering • Technology • Creativity

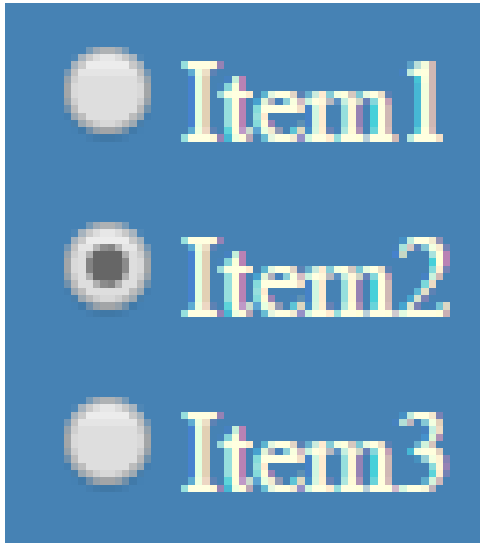
SWITH STATEMENT

```
protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    string myList = RadioButtonList1.SelectedItem.Text;

    switch (myList)
    {
        case ("Item1"):
            RadioButtonList1.BackColor = System.Drawing.Color.LightSlateGray;
            RadioButtonList1.ForeColor = System.Drawing.Color.White;
            RadioButtonList1.BorderColor = System.Drawing.Color.Gray;
            break;

        case ("Item2"):
            RadioButtonList1.BackColor = System.Drawing.Color.SteelBlue;
            RadioButtonList1.ForeColor = System.Drawing.Color.LightYellow;
            RadioButtonList1.BorderColor = System.Drawing.Color.MediumBlue;
            break;

        default:
            RadioButtonList1.BackColor = System.Drawing.Color.Gold;
            RadioButtonList1.ForeColor = System.Drawing.Color.Magenta;
            RadioButtonList1.BorderColor = System.Drawing.Color.DarkOrange;
            break;
    }
}
```

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

SWITH STATEMENT





FOR LOOP

- For Loops statement, which allows code to be repeatedly executed.

- In ASP.NET programming sometimes you have to faces situation to repeat a task for several times or you have to repeat a task till you reach a condition, in these situations you can use loop statements to achieve your desired results.

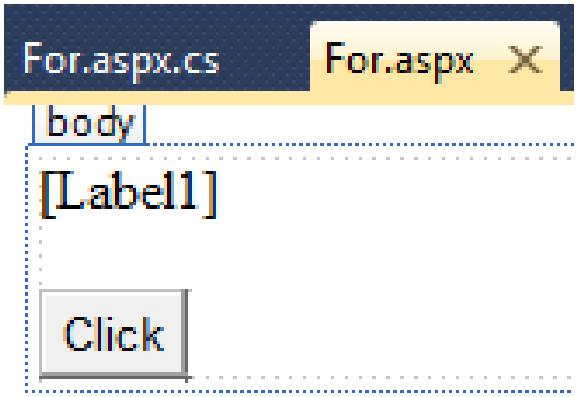
…FOR LOOP

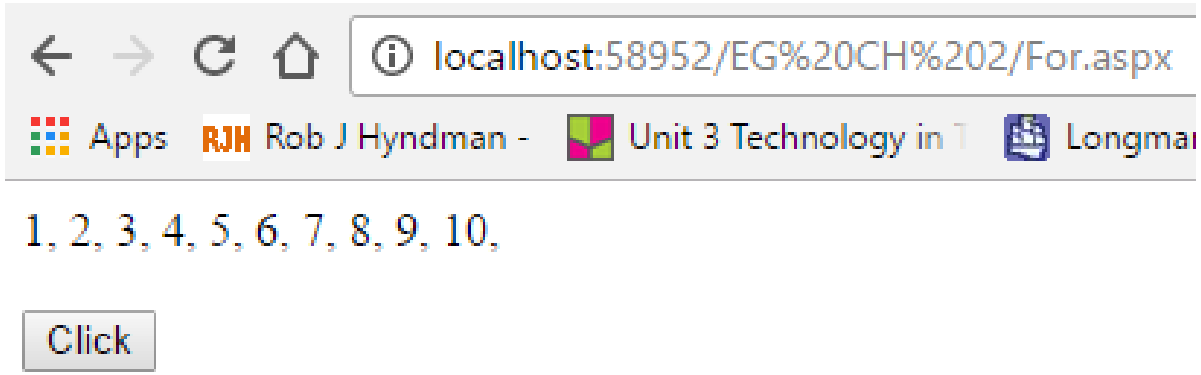
```
for (initialization; condition; step)

    statement
```

initialization : Initializing the value of variable
condition : Evaluate the condition
step : Step taken for each execution of loop body

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

...FOR LOOP






```
protected void Button1_Click(object sender, EventArgs e)
{
    int startVal = 1;
    int maxValue = 10;
    int i;

    for (i = startVal; i <= maxValue; i++)
    {
        Label1.Text += i + ", ";
    }
}
```

…FOR LOOP

```
int input = Convert.ToInt16(TBInput.Text);
int startVal = 1;
int i;

for (i = startVal; i <= input; i++)
{
    Label2.Text += i + ", ";
}
```

Eg 2:

1, 2, 3, 4, 5,

…FOR LOOP

```
protected void Button3_Click(object sender, EventArgs e)
{
    int i;

    string[] Colors = { "Red", "Pink", "Green", "Yellow", "Gray" };

    for (i = 0; i < Colors.Length; i++)
    {
        Label3.Text += Colors[i] + ", ";
    }
}
```

Eg 3:

Red, Pink, Green, Yellow, Gray,

LOOPING CONTROL STRUCTURE

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

WHILE

```
protected void Button1_Click(object sender, EventArgs e)
{
    int i=1, maxVal = 10;

    while (i <= maxVal)
    {
        Label1.Text += i + ", ";
        i += 1;
    }

}
```

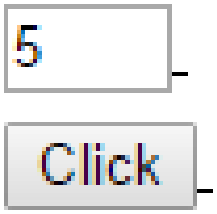
Eg 1:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

...WHILE

```
protected void Button2_Click(object sender, EventArgs e)
{
    int i = 1, input = Convert.ToInt16(TextBox1.Text);

    while (i<=input)
    {
        Label2.Text += i + ", ";
        i += 1;
    }
}
```

Eg 2:



1, 2, 3, 4, 5,

...WHILE

```
protected void Button3_Click(object sender, EventArgs e)
{
    int i=0;
    string[] colors = { "Fuschia", "DarkKhaki", "EmeraldGreen" };

    while (i< colors.Length)
    {
        Label3.Text += colors[i] + ", ";
        i += 1;
    }
}
```

Eg 3:

Fuschia, DarkKhaki, EmeraldGreen,

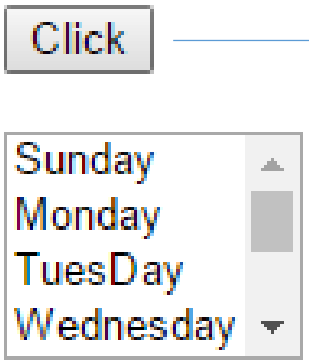
…DO WHILE

```
int i = 0;
string[] colors = { "Fuschia", "DarkKhaki", "EmeraldGreen" };

do
{
    Label1.Text += colors[i] + ", ";
    i += 1;
}
while (i < colors.Length);
```

***Don’t forget the semicolon at the end of do-while statement

FOREACH

```
protected void Button1_Click(object sender, EventArgs e)
{
    string[] days = { "Sunday", "Monday", "TuesDay", "Wednesday", "Thursday", "Friday", "Saturday" };
    ListBox1.Items.Clear();

    foreach (string day in days)
    {
        ListBox1.Items.Add(day);
    }
}
```



Click the button and all items in ‘days’ will be collected in the ListBox

FUNCTIONS

A function allows you to encapsulate a piece of code and call it from other parts of your code. You may very soon run into a situation where you need to repeat a piece of code, from multiple places, and this is where functions come in.

...FUNCTION

```
protected void Page_Load(object sender, EventArgs e)
{
    MultiplyNumbers(8,9);

    MultiplyNumbers(4,12); //Call function

    MultiplyNumbers(38,23);
}

void MultiplyNumbers(int A, int B) //Function name
{
    Label1.Text += A * B + "<br>";
}
```

72

48

874

...FUNCTION

```
public partial class Function : System.Web.UI.Page
{
    String getName()     //Function getName()
    {
        String name;

        name = TextBox1.Text;

        return name;     //return

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = getName();   //Call getame()
    }

}
```

Abdullah

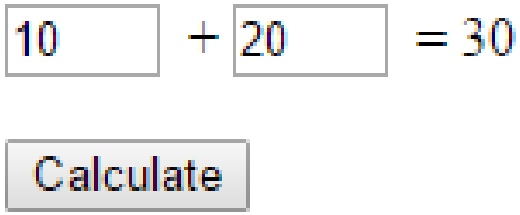
Click

Abdullah

*EG CH 2\Function.aspx.cs**

FUNCTION - EXERCISE

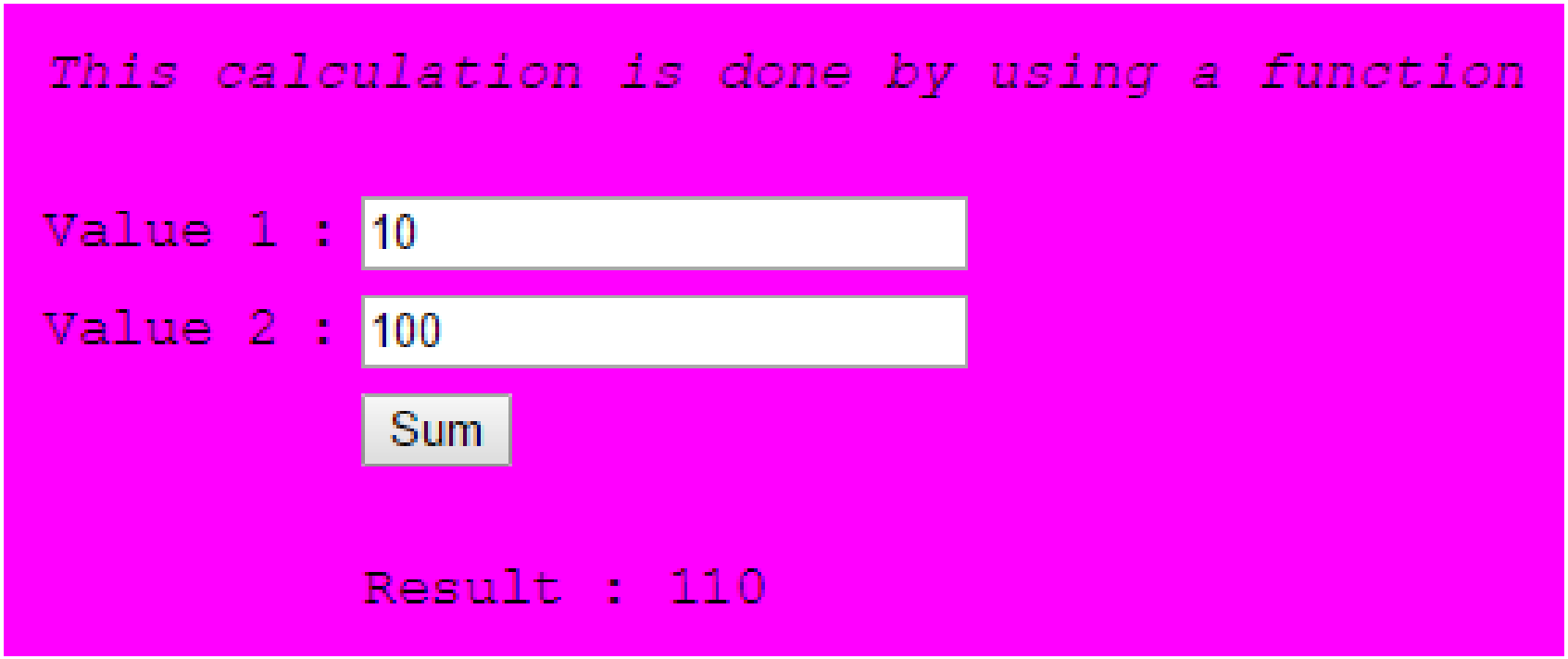
```
public partial class Function3 : System.Web.UI.Page
{
    int mySum()
    {
        int Value1 = Convert.ToInt16(TextBox1.Text);
        int Value2 = Convert.ToInt16(TextBox2.Text);
        int result;

        result = Value1 + Value2;

        if (result > 10)
        {
            return result;
        }

        return 0;
    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        Label1.Text = mySum().ToString();
    }
}
```



FUNCTION - EXERCISE

Based on the given code in “OPERATOR”, rewrite the code to a function.

Similar output should be obtained.



Function2.aspx.cs

FUNCTION

```
public partial class Function2 : System.Web.UI.Page
{
    int Calculate()
    {
        int Value1 = Convert.ToInt16(txtValue1.Text); //Convert the content in TextBox1 into an integer
        int Value2 = Convert.ToInt16(txtValue2.Text); //Convert the content in TextBox2 into an integer

        int Result = Value1 + Value2; //Do the calculation

        return Result;
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        lblResult.Text = "Result : " + Calculate().ToString();
    }
}
```

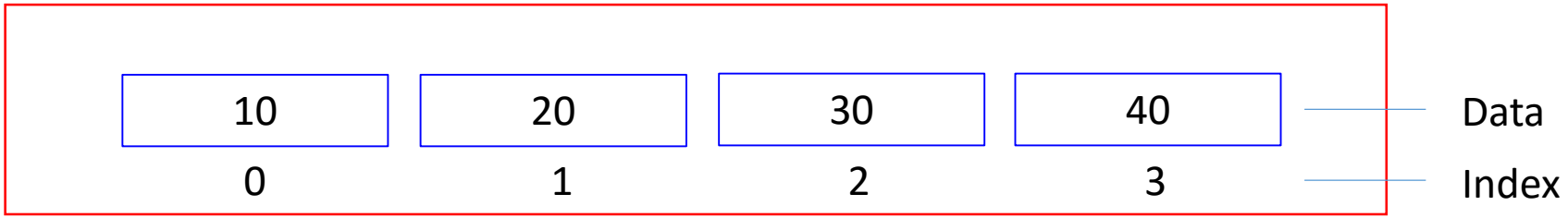
ARRAY

What is an array?

- Arrays are using to store similar data types grouping as a single unit.
- We can access Array elements by its numeric index.
- The array indexes start at zero.
- The default value of numeric array elements are set to zero, and reference elements are set to null.

DECLARE AN ARRAY

```
int [] array = new int {10, 20, 30, 40};
```



DECLARE AN ARRAY

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

Declaring and initializing an Integer Array.

```
int[] array = new int [4];

array [0] = 10;
Array [1] = 20;
array [2] = 30;
array [3] = 40;
```

- In the above code we declare an Integer Array of four elements and assign the value to array index . That means we assign values to array index 0 - 4.

- We can retrieve these values from array by using a for loop.

DECLARE AN ARRAY

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

Also we can declare and initialize an array in one statement.

```
int [] array = new int [] {10, 20, 30, 40};
```

Note that in the above code we did not specify the length of the array so the compiler will do it for us.

DECLARE AN ARRAY

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

String Array

Declaring and Initializing a String Array

In the following example, we declare an Array "week" capable of seven String values and assigns the seven values as days in a week . Next step is to retrieve the elements of the Array using a for loop . For finding the end of an Array we used the Length function of Array Object.

DECLARE AN ARRAY

...String Array

```
protected void Button1_Click(object sender, EventArgs e)
{
    String[] day = new string[7];
    day[0] = "Sunday";
    day[1] = "Monday";
    day[2] = "Tuesday";
    day[3] = "Wednesday";
    day[4] = "Thursday";
    day[5] = "Friday";
    day[6] = "Saturday";

    for (int i = 0; i < day.Length; i++)
    {
        Label1.Text += day[i] +"<br>";
    }
 }
```

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

Click

DECLARE AN ARRAY

- Another Example:

  First, we declare that `drinkList` is an array by following the datatype with two empty square brackets.

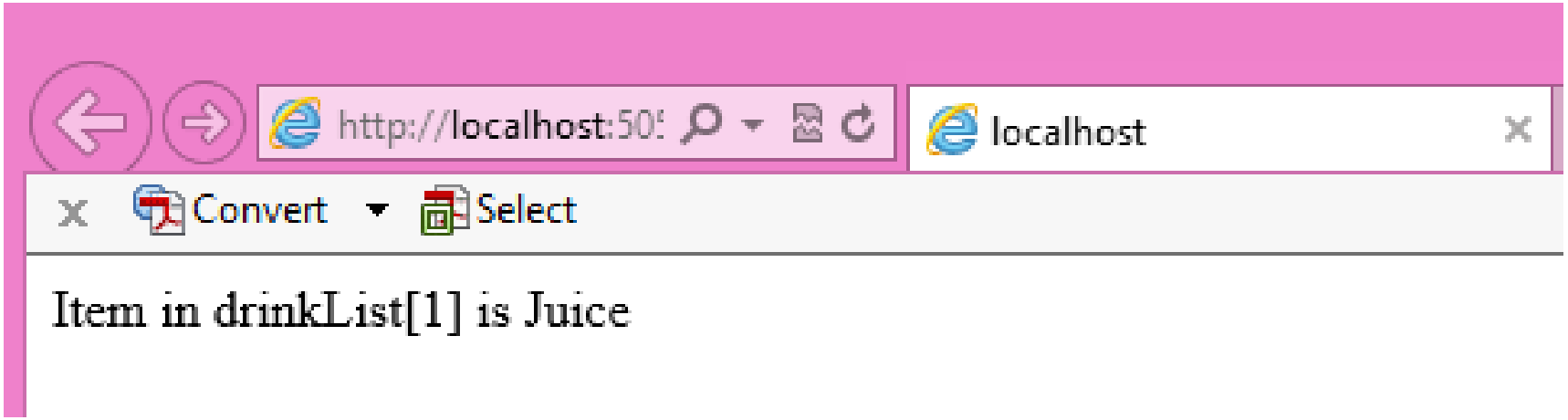
  We must then specify that this is an array of four items, using the `new` keyword:

```
protected void Page_Load(object sender, EventArgs e)
{
    String[] drinkList = new String[4];  //Declare an array

    drinkList[0] = "Water";
    drinkList[1] = "Juice";
    drinkList[2] = "Soda";
    drinkList[3] = "Milk";

    Label1.Text = "Item in drinkList[1] is " + drinkList[1];
}
```

//Place some items in it

//This line accesses an item in the array by its position

ARRAY


http://localhost:50
localhost
Convert
Select
Item in drinkList[1] is Juice

ARRAY

How to find the length of an Array ?

Array.Length

- We can use array.Length to find the length of an Array

So that's what an array is - **a variable that can hold more than one piece of data at a time**

TO BE

CONTINUED