|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dhairya Vyas |  | |  | | --- | | **Dot Net Technology – 2160711**  **Lab Manual**  Semester 6th  Academic Year 2019-20 Even  12th December 2019 – 14th April 2020 | |  | |



Gujarat Technological University

VVP Engineering College, Rajkot

Table of Contents

1. [Creating a hello World program using C# 3](#_Toc41835364)
2. [Generating various patterns in C# 4](#_Toc41835365)
3. [Getting input from user in C# 8](#_Toc41835366)
4. [INTFIX TO POSTFIX 10](#_Toc41835367)
5. [CONVERSIONs 12](#_Toc41835368)
6. [digits to words 15](#_Toc41835369)
7. [Currency Conversion 19](#_Toc41835370)
8. [Temperature conversion 21](#_Toc41835371)
9. [Change Font size 23](#_Toc41835372)
10. [Check Empty String 26](#_Toc41835373)
11. [Change colour of Label 32](#_Toc41835374)
12. [Textbox configuration 37](#_Toc41835375)
13. [Session State in SQL Server 40](#_Toc41835376)

**Practical 1**

# Creating a hello World program using C#

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello World !!");

Console.Read();

}

}

}

**Practical 2**

# Generating various patterns in C#

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical2 // This practical prints different patterns

{

class Program

{

/\* \* This pattern prints

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\*/

public static void Pattern1()

{

for(int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

Console.Write(" \*");

Console.WriteLine();

}

}

/\* \* This pattern prints

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\*/

public static void Pattern2()

{

for(int i = 0; i < 5; i++)

{

for (int j = 0; j <= i; j++)

Console.Write(" \*");

Console.WriteLine();

}

}

/\* \* This pattern prints

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*/

public static void Pattern3()

{

for (int i = 0; i < 5; i++)

{

for (int k = 4; k > i; k--)

Console.Write(" ");

for (int j = 0; j <= i; j++)

Console.Write(" \*");

Console.WriteLine();

}

}

/\* \* This pattern prints

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\*/

public static void Pattern4()

{

for (int i = 0; i < 5; i++)

{

for (int k = 4; k > i; k--)

Console.Write(" ");

for (int j = 0; j <= i; j++)

Console.Write(" \*");

Console.WriteLine();

}

}

/\* \* This pattern prints

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*/

public static void Pattern5()

{

for (int i = 0; i < 5; i++)

{

for (int k = 0; k < i; k++)

Console.Write(" ");

for (int j = 4-i; j >= 0; j--)

Console.Write(" \*");

Console.WriteLine();

}

}

/\* \* This pattern prints

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*/

public static void Pattern6()

{

for(int i = 0; i < 9; i++)

{

if (i < 5)

{

for (int k = 4; k > i; k--)

Console.Write(" ");

for (int j = 0; j <= i; j++)

Console.Write(" \*");

Console.WriteLine();

}

else

{

for (int k = 0; k < i-4; k++)

Console.Write(" ");

for (int j = 8 - i; j >= 0; j--)

Console.Write(" \*");

Console.WriteLine();

}

}

}

static void Main(string[] args)

{

Console.WriteLine("\tPattern 1\n");

Pattern1();

Console.WriteLine("\tPattern 2\n");

Pattern2();

Console.WriteLine("\tPattern 3\n");

Pattern3();

Console.WriteLine("\tPattern 4\n");

Pattern4();

Console.WriteLine("\tPattern 5\n");

Pattern5();

Console.WriteLine("\tPattern 6\n");

Pattern6();

Console.Read();

}

}

}

**Output:**



**Practical 3**

# Getting input from user in C#

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

// Simple Practical to take name and country as a input from user and display it after validating it

namespace Practical3

{

class Program

{

static void Main(string[] args)

{

string name, country;

Regex regex = new Regex(@"^[a-z]+$", RegexOptions.IgnoreCase);

// In this ary ^ stands for "must start with", + stands for 1 or more chars, $ stands for must end with

Console.WriteLine("!!! Warning :- Entered Name and Countery must be alphabets only else you will have to enter again.\n");

do {

Console.Write("Enter Your Name : ");

name = Console.ReadLine();

} while (regex.IsMatch(name) == false);

do

{

Console.Write("Enter Your Country name : ");

country = Console.ReadLine();

} while (regex.IsMatch(country) == false);

Console.WriteLine("Hey " + name + ",From " + country + ".");

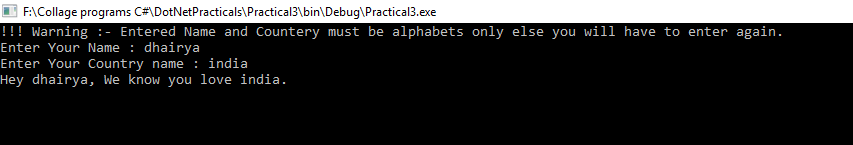
Console.Read();

}

}

}

**OUTPUT:-**



**Practical 4**

# INTFIX TO POSTFIX

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical4

{

class Program

{

bool IsAlNum(char c)

{

// Console.WriteLine(c);

if (c >= 'a' && c <= 'z')

return true;

if (c >= 'A' && c <= 'Z')

return true;

return false;

}

int Priority(char c)

{

if (c == '^' || c == '$')

return 3;

if (c == '%' || c == '\*' || c == '/')

return 2;

if (c == '+' || c == '-')

return 1;

return 0;

}

string InfixToPostfix(string infix)

{

Stack<char> s = new Stack<char>();

string postfix = "";

for (int i = 0; i < infix.Length; i++)

{

char ch = infix[i];

if(!IsAlNum(ch))

{

if (s.Count == 0)

{

s.Push(ch);

continue;

}

while(s.Count !=0 && Priority(s.Peek()) >= Priority(ch))

postfix += s.Pop();

s.Push(ch);

}

else

postfix += ch;

}

while (s.Count != 0)

postfix += s.Pop();

return postfix;

}

static void Main(string[] args)

{

string infix;

Console.Write("Enter Infix String : ");

infix = Console.ReadLine();

Program p = new Program();

string postfix = p.InfixToPostfix(infix);

Console.Write("Postifix is : " + postfix);

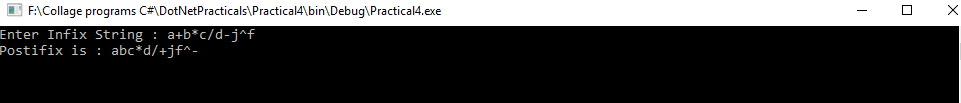
Console.Read();

}

}

}

***Output:***



**Practical 5\_6**

# CONVERSIONs

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical5\_6

{

class Program

{

public static string BinaryToDecimal(string Binary)

{

int Decimal = 0, temp, i = 0;

int binaryValue = int.Parse(Binary);

while (binaryValue > 0)

{

temp = (binaryValue % 10);

Decimal = Decimal + (temp \* (int)Math.Pow(2, i));

i++;

binaryValue = binaryValue / 10;

}

string Result = $"Binary: {Binary} is converted to Decimal: {Decimal}";

return Result;

}

public static string DecimalToBinary(int decimalValue)

{

string Result = $"Decimal: {decimalValue} is converted to Binary: ";

int value = decimalValue; int i;

int[] binaryString = new int[10];

for (i = 0; value > 0; i++)

{

binaryString[i] = (value % 2);

value = value / 2;

}

i--;

for (; i >= 0; i--)

{

Result += binaryString[i].ToString();

}

return Result;

}

public static string DecimalToOctal(int decimalValue)

{

string Result = $"Decimal: {decimalValue} is converted to Octal: ";

int value = decimalValue; int i;

int[] octalString = new int[10];

for (i = 0; value > 0; i++)

{

octalString[i] = (value % 8);

value = value / 8;

}

i--;

for (; i >= 0; i--)

{

Result += octalString[i].ToString();

}

return Result;

}

public static string DecimalToHexadecimal(int decimalValue)

{

string Result = $"Decimal: {decimalValue} is converted to Hexadecimal: ";

int value = decimalValue; int i;

int[] hexadecimalString = new int[10];

for (i = 0; value > 0; i++)

{

hexadecimalString[i] = (value % 16);

value /= 16;

}

i--;

for (; i >= 0; i--)

{

switch (hexadecimalString[i])

{

case 10: Result += "A"; break;

case 11: Result += "B"; break;

case 12: Result += "C"; break;

case 13: Result += "D"; break;

case 14: Result += "E"; break;

case 15: Result += "F"; break;

default: Result += hexadecimalString[i].ToString(); break;

}

}

return Result;

}

static void Main(string[] args)

{

Console.WriteLine(BinaryToDecimal("0101"));

Console.WriteLine(DecimalToBinary(31));

Console.WriteLine(DecimalToOctal(32));

Console.WriteLine(DecimalToHexadecimal(16));

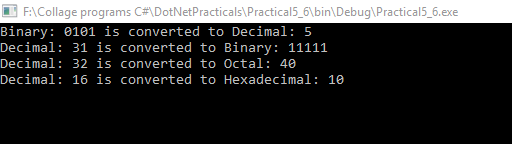
Console.ReadLine();

}

}

}

***Output:***



**Practical 7**

# digits to words

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical7

{

class Program

{

static string numberToWords(int number)

{

string words = "";

switch (number)

{

case 0: words = "zero"; break;

case 1: words = "one"; break;

case 2: words = "two"; break;

case 3: words = "three"; break;

case 4: words = "four"; break;

case 5: words = "five"; break;

case 6: words = "six"; break;

case 7: words = "seven"; break;

case 8: words = "eight"; break;

case 9: words = "nine"; break;

case 10: words = "ten"; break;

case 11: words = "eleven"; break;

case 12: words = "twelve"; break;

case 13: words = "thirteen"; break;

case 14: words = "fourteen"; break;

case 15: words = "fifteen"; break;

case 16: words = "sixteen"; break;

case 17: words = "seventeen"; break;

case 18: words = "eighteen"; break;

case 19: words = "nineteen"; break;

case 20: words = "twenty"; break;

case 30: words = "thirty"; break;

case 40: words = "forty"; break;

case 50: words = "fifty"; break;

case 60: words = "sixty"; break;

case 70: words = "seventy"; break;

case 80: words = "eighty"; break;

case 90: words = "ninety"; break;

case 100: words = "hundred"; break; //number > 100

case 1000: words = "thousand"; break; //number > 1000

case 1000000: words = "million"; break; //number > 1000

}

return words;

}

static int numberOfDigits(int number)

{

return (int)(Math.Log10(number) + 1);

}

static void Main(string[] args)

{

int number = 5678, i = numberOfDigits(number);

Console.Write(number + " in words :: ");

while (number > 0)

{

int tempNumber;

if (number % Math.Pow(10, i) > Math.Pow(10, i - 1))

{

tempNumber = number % (int)Math.Pow(10, i);

if (tempNumber > 100)

{

Console.Write(numberToWords(tempNumber / (int)Math.Pow(10, i - 1)) + " ");

Console.Write(numberToWords((int)Math.Pow(10, i - 1)) + " ");

}

else

{

if (tempNumber > 19)

Console.Write(numberToWords(tempNumber / 10 \* 10) + " ");

else

{

Console.Write(numberToWords(tempNumber) + " ");

}

}

}

number = number % (int)Math.Pow(10, i - 1);

i--;

}

Console.ReadLine();

}

}

}

***Output:***



**Practical 8**

# Currency Conversion

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Globalization;

using System.Threading.Tasks;

namespace Practical8

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter amount in Rupee :: ");

decimal Rupee, Dollar, Franc, Euro;

Rupee = decimal.Parse(Console.ReadLine());

Dollar = Rupee \* 0.0135107M;

Franc = Rupee \* 0.0126123M;

Euro = Rupee \* 0.0119623M;

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.WriteLine($"Dollar:{Dollar.ToString("C", new CultureInfo("en-US"))}");

Console.WriteLine($"Frank: {Franc.ToString("C", new CultureInfo("de-CH"))}");

Console.WriteLine($"Euro: {Euro.ToString("C", new CultureInfo("es-ES"))}");

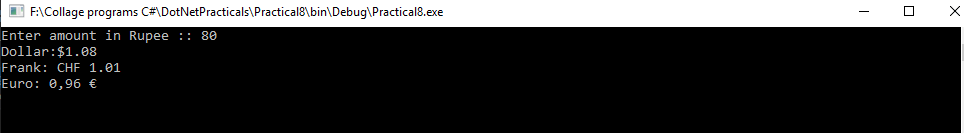
Console.ReadLine();

}

}

}

***Output:***



**Practical 9**

# Temperature conversion

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practical9

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Temperature Converter\n");

Console.WriteLine("1. To convert Celsius to Fahrenheit");

Console.WriteLine("2. To convert Fahrenheit to Celsius\n");

Console.Write("Enter your choice :: ");

int choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.Write("Enter a value in Celsius :: ");

float Celsius = float.Parse(Console.ReadLine());

float Fahrenheit = Celsius \* (9.0f / 5.0f) + 32;

Console.WriteLine($"Fahrenheit :: {Fahrenheit}");

break;

case 2:

Console.Write("Enter a value in Fahrenheit :: ");

Fahrenheit = float.Parse(Console.ReadLine());

Celsius = (Fahrenheit - 32) \* (5.0f / 9.0f);

Console.WriteLine($"Celsius :: {Celsius}");

break;

default:

Console.WriteLine("Invalid Option Selected.");

break;

}

Console.ReadLine();

}

}

}

***Output:***



**Practical 9**

# Change Font size

***Program.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practical10

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

***Form.cs:***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practical10

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void buttonIncrease\_Click(object sender, EventArgs e)

{

labelText.Font = new Font(labelText.Font.Name, labelText.Font.Size + 3);

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void buttonDecrease\_Click(object sender, EventArgs e)

{

if (labelText.Font.Size - 3 <= 0)

{

MessageBox.Show("Bas pan have ketlu hoyyy !!!!");

}

else

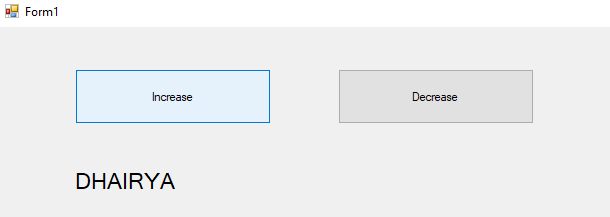
labelText.Font = new Font(labelText.Font.Name, labelText.Font.Size - 3);

}

}

}

***Output:***



**Practical 11**

# Check Empty String

***WebForm1.aspx:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical11

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Request.QueryString.HasKeys())

{

lblUsername.Text = Request.QueryString["username"];

}

else

{

lblUsername.Text = "Empty QueryString";

}

}

}

}

***WebForm1.aspx.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical11

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Request.QueryString.HasKeys())

{

lblUsername.Text = Request.QueryString["username"];

}

else

{

lblUsername.Text = "Empty QueryString";

}

}

}

}

***WebForm1.aspx.designer.cs:***

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Practical11

{

public partial class WebForm1

{

/// <summary>

/// form1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.HtmlControls.HtmlForm form1;

/// <summary>

/// lblUsername control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Label lblUsername;

}

}

***WebForm2.aspx:***

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="CheckEmptyString.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<table style="width:100%;">

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>

<asp:Literal ID="Literal1" runat="server" Text="Username"></asp:Literal>

</td>

<td>

<asp:TextBox ID="txtUsername" runat="server"></asp:TextBox>

</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>

<asp:Literal ID="Literal2" runat="server" Text="Password"></asp:Literal>

</td>

<td>

<asp:TextBox ID="txtPassword" runat="server"></asp:TextBox>

</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>

<asp:Button ID="btnLogin" runat="server" OnClick="btnLogin\_Click" Text="Login" />

</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

</div>

</form>

</body>

</html>

***WebForm2.aspx.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical11

{

public partial class WebForm2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnLogin\_Click(object sender, EventArgs e)

{

string queryString = “”;

Response.Redirect("WebForm1.aspx?" + queryString);

}

}

}

***WebForm2.aspx.designer.cs:***

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Practical11

{

public partial class WebForm2

{

/// <summary>

/// form1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.HtmlControls.HtmlForm form1;

/// <summary>

/// Literal1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Literal Literal1;

/// <summary>

/// txtUsername control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.TextBox txtUsername;

/// <summary>

/// Literal2 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Literal Literal2;

/// <summary>

/// txtPassword control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.TextBox txtPassword;

/// <summary>

/// btnLogin control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button btnLogin;

}

}

***Output:***

******

******

**Practical 12**

# Change colour of Label

***WebForm1.aspx:***

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="ChangeColourLabel.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

Change the color of this label:

<asp:Label ID="lblChangingColor" runat="server" Font-Bold="True" Font-Size="30pt" Text="Label"></asp:Label><br/><br/>

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Change Color" />

</div>

<asp:Panel ID="Panel1" runat="server">

<asp:Button ID="Button2" runat="server" BackColor="Yellow" Height="50px" OnClick="Button2\_Click" Width="50px" />

<asp:Button ID="Button3" runat="server" BackColor="Red" Height="50px" OnClick="Button3\_Click" Width="50px" />

<asp:Button ID="Button4" runat="server" BackColor="#0033CC" Height="50px" OnClick="Button4\_Click" Width="50px" />

&nbsp;

<asp:Button ID="btnSetColor" runat="server" OnClick="btnSetColor\_Click" Text="Select" />

</asp:Panel>

</form>

</body>

</html>

***WebForm1.aspx.cs:***

using System;

using System.Collections.Generic;

using System.Drawing.Configuration;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical12

{

public partial class WebForm1 : System.Web.UI.Page

{

static System.Drawing.Color color;

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

Panel1.Visible = false;

}

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (Panel1.Visible == true)

{

Panel1.Visible = false;

}

else

{

Panel1.Visible = true;

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

color = System.Drawing.Color.Yellow;

}

protected void Button3\_Click(object sender, EventArgs e)

{

color = System.Drawing.Color.Red;

}

protected void Button4\_Click(object sender, EventArgs e)

{

color = System.Drawing.Color.Blue;

}

protected void btnSetColor\_Click(object sender, EventArgs e)

{

lblChangingColor.ForeColor = color;

Panel1.Visible = false;

}

}

}

***WebForm1.aspx.designer.cs:***

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Practical12

{

public partial class WebForm1

{

/// <summary>

/// form1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.HtmlControls.HtmlForm form1;

/// <summary>

/// lblChangingColor control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Label lblChangingColor;

/// <summary>

/// Button1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button Button1;

/// <summary>

/// Panel1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Panel Panel1;

/// <summary>

/// Button2 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button Button2;

/// <summary>

/// Button3 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button Button3;

/// <summary>

/// Button4 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button Button4;

/// <summary>

/// btnSetColor control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

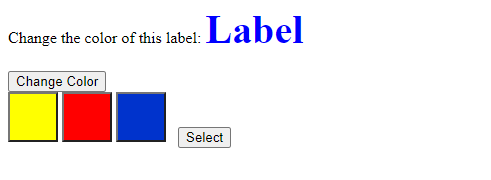
/// </remarks>

protected global::System.Web.UI.WebControls.Button btnSetColor;

}

}

***Output:***

******

**Practical 13**

# Textbox configuration

***WebForm1.aspx:***

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="TextBoxConfig.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

You can enable and disable this textbox:

<asp:TextBox ID="txtEnableDisable" runat="server"></asp:TextBox>

<br />

<asp:Button ID="btnEnableDisable" runat="server" OnClick="btnEnableDisable\_Click" Text="Enable/Disable" />

<br />

<asp:TextBox ID="TextBox1" runat="server" Width="117px"></asp:TextBox>

<asp:Button ID="btnChangeWidth" runat="server" OnClick="btnChangeWidth\_Click" Text="Change Width" />

</div>

</form>

</body>

</html>

***WebForm1.aspx.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical13

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnEnableDisable\_Click(object sender, EventArgs e)

{

if (txtEnableDisable.Enabled ==true)

{

txtEnableDisable.Enabled = false;

}

else

{

txtEnableDisable.Enabled = true;

}

}

protected void btnChangeWidth\_Click(object sender, EventArgs e)

{

txtEnableDisable.Width =int.Parse(TextBox1.Text);

}

}

}

***WebForm1.aspx.designer.cs:***

namespace Practical13

{

public partial class WebForm1

{

protected global::System.Web.UI.HtmlControls.HtmlForm form1;

protected global::System.Web.UI.WebControls.TextBox txtEnableDisable;

protected global::System.Web.UI.WebControls.Button btnEnableDisable;

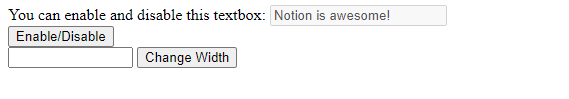
protected global::System.Web.UI.WebControls.TextBox TextBox1;

protected global::System.Web.UI.WebControls.Button btnChangeWidth;

}

}

***Output:***

******

******

**Practical 14**

# Session State in SQL Server

***WebForm1.aspx:***

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="SessionState.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<table>

<tr>

<td colspan="2"><h1>Web Page 1</h1></td>

</tr>

<tr>

<td>Name:</td>

<td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td>Mobile:</td>

<td><asp:TextBox ID="txtMobile" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td colspan="2"><asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click" /></td>

</tr>

</table>

</div>

</form>

</body>

</html>

***WebForm1.aspx.cs:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical14

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

Session["Name"] = txtName.Text;

Session["Mobile"] = txtMobile.Text;

Response.Redirect("WebForm2.aspx");

}

}

}

***WebForm1.aspx.designer.cs:***

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated by a tool.

//

// Changes to this file may cause incorrect behavior and will be lost if

// the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace Practical14

{

public partial class WebForm1

{

/// <summary>

/// form1 control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.HtmlControls.HtmlForm form1;

/// <summary>

/// txtName control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.TextBox txtName;

/// <summary>

/// txtMobile control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.TextBox txtMobile;

/// <summary>

/// btnSubmit control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button btnSubmit;

}

}

***Output:***

******