# Part A: Ex No: 1- Accept a character and check the case of Character

## Aim:

Accept a character from console and check the case of the character.

## Pre-Learning Skills

- Knowing the templates installed and version of visual studio used
- Understanding the different types of application in visual studio
- How to save the application in their own directories
- Identifying the commands to build and run the solutions

## Procedure:

1. Open a new Project.
2. Select the console application from the templates and locate the path to store it.
3. Print title or heading for the program
4. Accept a character using ReadLine () function.
5. Check for the character case by its ASCII value.
6. Save and execute the code

## Source Code: LP_1 – To Change Case using built in methods

```
using System;

namespace Lab_Practice_1_1
{
   internal class LP1_1
   {
       static void Main(string[] args)
       {
           string input_str;
           char input_char;

           Console.WriteLine("Enter a single charecter:");
           input_str = Console.ReadLine();         // Read String from user

           if (input_str.Length != 1)               // Check if exactly 1 char entered
            {
                Console.WriteLine("ERROR: Enter exactly 1 charecter only.");
            }
            else
            {
                input_char = char.Parse(input_str);     // Convert string to char

                Console.WriteLine("Given character is: " + input_char);

                if (Char.IsLower(input_char))
                    Console.WriteLine("The entered character is Lowercase.");

                else if (Char.IsUpper(input_char))
                    Console.WriteLine("The entered character is Uppercase.");

                else
                    Console.WriteLine("The entered character is not an alphabet.");
            }

            Console.WriteLine("Press any key to continue . . .");
            Console.ReadKey();
        }
    }
}
```

## **Source Code: LP1_2 –** To Change Case using User Defined Function

```
using System;

namespace LP1_2
{
   class Program
   {
       static void Main(string[] args)
       {
          Console.Write("\n\nProgram to Check whether a character is lower case or
           upper case :\n");
          Console.Write("--------------------------------------------------------\n");
          Console.Write("Input a Single character : ");
          char ch = (char)Console.Read();
          // Call of Function
          checkCase(ch);
          Console.ReadKey();
        }
        //Function Definition
        static void checkCase(char ch)
        {
            if (ch >= 'A' && ch <= 'Z')
                Console.WriteLine("\n" + ch +
                     " is an UpperCase character");
            else if (ch >= 'a' && ch <= 'z')
                Console.WriteLine("\n" + ch +
                     " is an LowerCase character");
        }
    }
}

```

## Source Code: Ex_1

```
using System;

namespace Ex_1
{
   class Program
   {
       static void Main(string[] args)
       {
           Console.Write("Program to check whether a character is lower or upper case.\n");
           Console.Write("------------------------------------------------\n");
           Console.Write("Input a character: ");
           char ch = (char)Console.Read();

           if (Char.IsUpper(ch))
           {
               Console.WriteLine("\nThe character is uppercase.\n");
            }
            else if(Char.IsLower(ch))
            {
               Console.WriteLine("\nThe character is lowercase.\n");
            }
            else
            {
  // User has entered a symbol or number etc. Example: 1,2, #, ?...
                Console.WriteLine("\nThe character is not an alphabet.\n");
            }
            Console.ReadKey();
        }
    }
}
```

## Explanation:

1. ReadLine ( ) function works similar to scanf ( ) function. Waits for the user until an input is given from the keyboard
2. Write ( ) and WriteLine ( ) functions work similar to printf ( ) function
3. Readkey ( ) accepts one character similar to getch( ) function

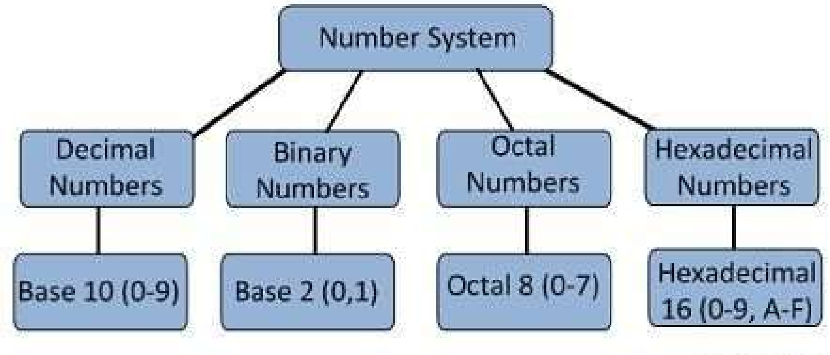
**Output:** (Write down the screen output here)

## Result:

The above program ______________________ was successfully executed and then the output was verified under ______________________.

## Viva Questions

1. List out the decision making system
   - Simple if statement
   - if … else
   - else…if ladder
   - switch…case
2. Number System



3. What is ASCII value?

   The ASCII decimal (Dec) number is created from binary, which is the language of all computers. For an example, the lower case "h" character (char) has a decimal value of 104, which is "01101000" in binary.

4. **How will you read numeric data using Readline( )?**

   ReadLine ( ) function returns a 'string', so in case we want to work with an integer number we have to convert the string to integer by using Convert.ToInt16 (string).

   **Example:**

```
Console.Write("Please Enter your age: ");
Age = Convert.ToInt16(Console.ReadLine());
Console.WriteLine("User Age: " + Age);
```

## Skill Practice;

- Write down the American Standard Code for Information Interchange(ASCII) value for the string HELLO

| Letter | Decimal Value(ASCII) | Binary Value for the letter |
|---|---|---|
| H | | |
| E | | |
| L | | |
| L | | |
| O | | |

# Part A: Ex No: 2 - Accept any character and Display Vowel or Not

## Aim:

Develop a Vb.Net program to accept any one character from keyboard and display whether it is vowel or not.

## Pre-Learning Skills

- Understanding the vowels and consonants
- Must know the Relational and Logical Operators available

## Procedure:

1. Open a new Project.
2. Select the console application from the templates and locate the path to store it.
3. Accept a character using ReadLine () function.
4. Check for the character if it belongs to,' A', 'E','I','O','U'.
5. Save and execute the code.

## Source Code: LP_2_1

```
using System;
namespace LP2_1
{
   class Program
   {
       static void Main(string[] args)
       {
           string input_str;
           char input_char;
            Console.WriteLine("Enter a Single Character");
            // Reads an input from user
            input_str = (Console.ReadLine());
            // Checks if user entered input is 1 character
            if (input_str.Length != 1)
                Console.WriteLine("Given Input is  " + input_str);
            else
            {
                // Converts string to a Single Character
                input_char = char.Parse(input_str);
                Console.WriteLine("Given Character is  " +
                input_char);
                // Condition for vowel checking
                if (input_char == 'a' || input_char == 'e' ||
                   input_char == 'i' || input_char == 'o' ||
                   input_char == 'u' || input_char == 'A' ||
                   input_char == 'E' || input_char == 'I' ||
                   input_char == 'O' || input_char == 'U')
                   Console.WriteLine("Entered Character '" +
                   input_char + "' is Vowel.");
```

```
                    else
                        Console.WriteLine("Entered Character '" +
                        input_char + "' is Not a Vowel.");
                }
                Console.ReadKey();
            }
        }
    }

```

## Source Code: LP_2_2

```
using System;

namespace LP2_2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter an Alphabet");
            char ch = Convert.ToChar(Console.ReadLine().ToLower());
            switch (ch)
            {
                case 'a':
                    Console.WriteLine("It is vowel");
                    break;
                case 'i':
                    Console.WriteLine("It is vowel");
                    break;
                case 'o':
                    Console.WriteLine("It is vowel");
                    break;
                case 'u':
                    Console.WriteLine("It is vowel");
                    break;
                case 'e':
                    Console.WriteLine("It is vowel");
                    break;
                default:
                    Console.WriteLine("It Is Not Vowel");
                    break;
            }
            Console.ReadKey();
        }
    }
}
```

## Source Code: LP2_3

```
using System;

namespace LP2_3
{
    class Program
   {
       static void Main(string[] args)
       {
           string vowels = "aeiouAEIOU";
            // get char input from user
            Console.WriteLine("Enter One Character");
            // Call of Function
            string char_to_check = get_one_char();
            //Print given caharacter
            Console.WriteLine(char_to_check);
            // Checks if character is in string
            if (vowels.Contains(char_to_check))
                Console.WriteLine("Given Character is Vowel");
            else
                Console.WriteLine("Given Character is Not Vowel");

            Console.ReadKey();
        }
        static string get_one_char()  // Function definition
        {
            string inp_str = Console.ReadLine();
            char c = inp_str[0];
            return (inp_str);
        }
    }
}

```

## Source Code: Ex_2

```
using System;

namespace Ex_2
{
    class Program
    {
        static void Main(string[] args)
        {
            string vowels = "aeiouAIEOU";
            string input_str;

            Console.WriteLine("Program to check if charecter is Vowel.");
            Console.WriteLine("Enter a Single Character:");
             // Reads input string
            input_str = Console.ReadLine();

            // Check if string has single charecter only
            if (input_str.Length == 1)
            {
                Console.WriteLine("Given Character is  " + input_str);
                // check if given charecter is present in "aeiouAEIOU" string
                if (vowels.Contains(input_str))
                    Console.WriteLine(" Entered Character is Vowel.");
                else
                    Console.WriteLine(" Entered Charecter is not vowel.");
            }
            else
                Console.WriteLine("ERROR:Enter 1 charecter only.");

            Console.WriteLine("Press any key...");
            Console.ReadKey();
        }
    }
}
```

**Output:** (Write down the screen output here)

## Result:

The above program ______________________________ was successfully executed and then the output was verified under ______________________________.

## Viva Questions

**List out some of the various string operations and functions used in c#**

| String Functions | Definitions |
|---|---|
| `CompareTo()` | Compare two strings and returns integer value as output. It returns 0 for true and 1 for false. |
| `Contains()` | The C# Contains method checks whether specified character or string is exists or not in the string value. |
| `EndsWith()` | This EndsWith Method checks whether specified character is the last character of string or not. |
| `Equals()` | The Equals Method in C# compares two string and returns Boolean value as output. |
| `ToLower()` | Converts String into lower case based on rules of the current culture. |
| `ToUpper()` | Converts String into Upper case based on rules of the current culture. |
| `Length` | It is a string property that returns length of string. |
| `Substring()` | This method returns substring. |
| `Trim()` | It removes extra whitespaces from beginning and ending of string. |

# Part A: Ex No: 3 - To implement a Calculator with memory and recall operations.

## Aim:

To develop a Vb.Net program to implement a calculator with Memory and Recall operations.

## Pre-Learning Skills

- Understanding the Integrated Development Environment(IDE)
- Understanding the different input controls from the tool box
- Learning the basic mathematical formulas and arithmetic operations
- Usage of control flow statements (switch..case)

## Procedure:

1. Open a New project.
2. Choose the windows application from the Templates.
3. Design the Form as a calculator by drag and drop array of button control for numbers & operator symbol and textbox control for inputs
4. Change the Form Caption to the Title of the program
5. Drag and Drop required Controls from the tool box and change the name and text properties.
6. Place separate buttons for arithmetic and memory operations.
7. Write the following code in the click () event of the button.
8. Save and Execute.

## Source Code: LP_3_1- Simple Arithmetic Operations

```
using System;

namespace LP3_1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the action to be performed");
            Console.WriteLine("Press 1 for Addition");
            Console.WriteLine("Press 2 for Subtraction");
            Console.WriteLine("Press 3 for Multiplication");
            Console.WriteLine("Press 4 for Division \n");
            int action = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter 1st input");
            int input_1 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter 2nd input");
            int input_2 = Convert.ToInt32(Console.ReadLine());

            int result = 0;
            switch (action)
            {
                case 1:
                    {
                        result = Addition(input_1, input_2);
                        break;
                    }
                case 2:
                    {
                        result = Subtraction(input_1, input_2);
                        break;
                    }
                case 3:
                    {
                        result = Multiplication(input_1, input_2);
                        break;
                    }
                case 4:
                    {
                        result = Division(input_1, input_2);
                        break;
                    }
                default:
                    Console.WriteLine("Wrong action!! try again");
                    break;
            }
            Console.WriteLine("The result is {0}", result);
            Console.ReadKey();
        }
        // Addition
        public static int Addition(int input_1, int input_2)
        {
            int result = input_1 + input_2;
            return result;
        }
        // Substraction
        public static int Subtraction(int input_1, int input_2)
        {
            int result = input_1 + input_2;
            return result;
        }
        // Multiplication
        public static int Multiplication(int input_1, int input_2)
        {
            int result = input_1 + input_2;
            return result;
        }
        // Division
        public static int Division(int input_1, int input_2)
        {
            int result = input_1 + input_2;
            return result;
        }
    }
}
```

## Source Code: Ex_3

```
using System;
using System.Windows.Forms;

namespace Ex_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        double operand1,operand2;
        string operation = string.Empty;
        double memory, result = 0.0;

        private void NumEvent(object sender, EventArgs e)
        {
            Button btn = (Button)sender;
            txt_result.Text += btn.Text;
        }

        private void OperatorsEvent(object sender, EventArgs e)
        {
            Button btn = (Button)sender;
            operation = btn.Text;

            if (txt_result.Text != "")
                operand1 = double.Parse(txt_result.Text);
            else
                operand1 = 0;

            txt_result.Text = "";
        }

        private void btn_clear_Click(object sender, EventArgs e)
        {
            txt_result.Clear();
        }

        private void btn_memory_Click(object sender, EventArgs e)
        {
            if(txt_result.Text!="")
                memory += double.Parse(txt_result.Text);
        }

        private void btn_memory_clear_Click(object sender, EventArgs e)
        {
            memory = 0;
        }

        private void btn_recall_Click(object sender, EventArgs e)
        {
            txt_result.Text = memory.ToString();
        }

        private void btn_result_Click(object sender, EventArgs e)
        {
            if (txt_result.Text != "")
                operand2 = double.Parse(txt_result.Text);
            else
                operand2 = 0;

            switch (operation)
            {
                case "+":
                    result = (operand1) + (operand2);
                    break;
                case "-":
                    result = (operand1) - (operand2);
                    break;
                case "*":
                    result = (operand1) * (operand2);
                    break;
                case "/":
                    if (operand2 != 0)
                        result = (operand1) / (operand2);
                    else
                    {
                        MessageBox.Show("Cannot divide by zero.","Error");
                        result = 0;
                    }
                    break;
                default:
                    break;
            }

            txt_result.Text = result.ToString();
        }
    }
}
```

**Output:** (Print your screen for output)

**Result**:(Write down the result)

## Viva Questions

1. **What is the difference between "continue" and "break" statements in C#?**

   Using break statement, you can 'jump out of a loop' whereas by using continue statement, you can 'jump over one iteration' and then resume your loop execution.

## 2. Mathematical Function in c#

| Math Functions | Description |
|---|---|
| Abs | Absolute value. Abs (a) corresponds to a = a if a> = 0 and a = -a if a <0 |
| Sqrt | To find the square root |
| Pow | Pow (x,y) corresponds to $x^Y$ |
| Exp | Exp (x) corresponds to $e^X$. where e is the base of the natural is logarithms |
| Log | natural logarithm. Logarithm to base e |
| Log10 | Logarithm to base 10 |
| Sin | Sine value |
| Asin | Arc sine |
| Sinh | Hyperbolic sine |
| Cos | Cosine |
| Acos | Arc cosine |
| Cosh | Cosine hyperbolic |
| Tan | Tangent |
| Atan | Arctangent |
| Tanh | Hyperbolic tangent |
| Round | Round the floating point numbers |
| Ceiling | Round Up |
| Floor | Round off |
| Max | Max (a,b) is the larger of the values a and b |
| Min | Min (a ,b) is the smaller of the values a and b |

## Skill Practice;

- List out the windows from Visual studio Integrated Development Environment(IDE)

| | |
|---|---|
| | |
| | |
| | |
| | |

- List out the basic property name from the properties window

| Property Name | Purpose or Usage | Property Name | Purpose or Usage |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

# Part A: Ex No: 4 - To implement a Calendar control to pick a date and display the day

## Aim:

Develop a form in VB.Net to pick a date from calendar control and display the Day, Month, Year details in separate Textboxes.

## Pre-Learning Skills

- Understanding the date and time functions
- Identifying the various format of date available

## Procedure:

1. Open a New project.
2. Choose the **windows application** from the Templates.
3. Design the Form as follows
   - Change the Form Caption to the Title of the program
   - Drag and Drop required Controls from the tool box and change the name and text properties.
   - Add a Calendar control to select the date.
4. Write the following code in the click () event of all the buttons.
5. Save and Execute (press F5).

## Source Code: LP_4_1

```
using System;
using System.Windows.Forms;
namespace LP4_1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
         }
         private void button1_Click(object sender, EventArgs e)
         {
             // Get the current date in a datetime format i.e,
             //12/04/2021
                 string currentDate1 =
                 DateTime.Now.ToString("dd/MM/yyyy");
                // Displaying the current date
                MessageBox.Show(currentDate1);
         }
     }
}
```

## Source Code: LP_4_2

```
using System;
using System.Windows.Forms;

namespace LP4_2
{
   public partial class Form1 : Form
   {
       public Form1()
       {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            listBox1.Items.Add(textBox1.Text);
        }
        private void listBox1_SelectedValueChanged()
        {
            DateTime currentDate1 = DateTime.Now;
            label1.Text = currentDate1.ToString(listBox1.Text);
        }
    }
}
```

## Source Code: LP_4_3

```
using System;
using System.Windows.Forms;

namespace LP4_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btn_click_to_add_Click()
        {
            DateTime currentDate1 = DateTime.Now;
            DateTime addedDate =
            currentDate1.AddDays(int.Parse(txt_input.Text));
            MessageBox.Show(addedDate.ToString("F"));
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            txt_current_date.Text = DateTime.Now.ToString();
        }
        private void btn_add_month_Click()
        {
            DateTime currentDate1 = DateTime.Now;
            DateTime addedmonth =
            currentDate1.AddMonths(int.Parse(txt_input.Text));
            MessageBox.Show(addedmonth.ToString("F"));
        }
        private void btn_add_year_Click(object sender, EventArgs e)
        {
            DateTime currentDate1 = DateTime.Now;
            DateTime addedyear =
            currentDate1.AddYears(int.Parse(txt_input.Text));
            MessageBox.Show(addedyear.ToString("F"));
        }
    }
}

```

**Source Code: Ex_4**

```
using System;
using System.Windows.Forms;

namespace Ex_4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void dateTimePicker1_ValueChanged()
        {
            DateTime dt = dateTimePicker1.Value;
            //Displays the day number
            txt_day.Text = dt.Day.ToString();
            // Displays the month number
            txt_month.Text = dt.Month.ToString();
            // Displays the year
            txt_year.Text = dt.Year.ToString();
            //Displays the day
            txt_dayname.Text = dt.ToString("dddd");
            //Displays the month name
            txt_monthname.Text = dt.ToString("MMMM");
        }
    }
}

```

**Output:** (Print your screen for output)

**Result** :( Write down the result)

## Viva Questions

1. **How to increase the Date corresponding with month,date,year?**

   DateSerial(year(Now),Month(Now)+1,1)

   Hour, min, sec, month, year, DateSerial, dateadd, datediff, weekday, datevalue, timeserial,timevalue.

2. **Name some date function?**

   Dateadd (), Datediff (), Datepart (), Cdate ()

3. **What is Mask Edit and why it is used?**

   Masked intelligent user control enhances the function of the TextBox control, which can mask the Date, IP Address, Phone numbers, digits, decimal and checks the validation, and automatically set the delimiter location.

## Skill Practice;

❖ Design a window form using calendar control and to find your age as on today

# Part A: Ex No: 5 - To Implement Common Dialog Control with File and Directory Controls

## Aim:

To develop a VB.Net application using the File, Directory and Directory Controls to implement a Common dialog box.

## Pre-Learning Skills

- Understanding the list of common dialog controls from the tool box
- Properties of each control can be analyzed
- Learn about events and its types

## Procedure:

1. Open a New project.
2. Choose the **windows application** from the Templates.
3. Add the following .net components from library (right click toolbox and click choose items)
    - Drivelistbox
    - Dirlistbox
    - Filelistbox
4. Design the Form as follows
    - Drag and Drop required Controls from the tool box and change the name and text properties.
5. Write the following code in the click () event of all the buttons.
6. Save and Execute (press F5).

**Source Code: LP_5_1-** To display selected color using color dialog

```
using System;
using System.Windows.Forms;
namespace Lab_Practice_5_1
{
    public partial class LP_5_1 : Form
    {
            private void show_color_Click(object sender, EventArgs e)
        {
                    ColorDialog dlg = new ColorDialog();
                    dlg.ShowDialog();
```

```
                 if (dlg.ShowDialog() == DialogResult.OK)
                 {
                         string str = null;
                         str = dlg.Color.Name;
                     MessageBox.Show (“Selcted Color is ‟ + str);

                 }
                }
     }
 }

```

## Source Code: LP_5_2 - To Change the text font using font dialog

```
using System;
using System.Windows.Forms;
namespace Lab_Practice_5_2
{
    public partial class LP_5_2 : Form
    {

      private void show_color_Click(object sender, EventArgs e)
      {
          FontDialog fontdlg = new FontDialog();
          fontdlg.ShowDialog();
          if (fontDlg.ShowDialog() != DialogResult.Cancel)
           {
               textBox1.Font = fontDlg.Font;
               label1.Font = "Selected Font ‟ + fontDlg.Font;
             }
             }
    }
}

```

## Source Code: Ex_5

```
using System;
using System.Windows.Forms;

namespace Ex_5
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void fileListBox1_SelectedIndexChanged()
        {
            txt_foldername.Text = fileListBox1.Path;
            txt_foldername.Text = dirListBox1.Path;
            txt_filename.Text = (fileListBox1.FileName);
            // Loads the image in picture box
            pictureBox1.ImageLocation = dirListBox1.Path + "\\" +
            fileListBox1.FileName;
        }
        private void driveListBox1_SelectedIndexChanged()
        {
            dirListBox1.Path = driveListBox1.Drive;
        }
        private void dirListBox1_SelectedIndexChanged()
        {
            fileListBox1.Path = dirListBox1.Path;
        }
        private void btn_select_folder_Click()
        {
            folderBrowserDialog1.ShowDialog();
            txt_foldername.Text =
                        folderBrowserDialog1.SelectedPath;
        }
        private void btn_select_file_Click()
        {
            openFileDialog1.ShowDialog();
            txt_filename.Text = openFileDialog1.FileName;
        }
    }
}
```

**Output:** (Print your screen for output)

**Result:**(Write down the result)

## Viva Questions

1. **How to assign FONT to label during runtime?**

   Label.font.name=”Times New Roman”;

   Label.font.size=20

2. **What are common dialog controls?**

   The common dialog boxes control, which is a custom control, allows your project to use the dialog boxes that are provided as a part of the Windows environment.

3. **Name some showprinter method in Printer object?**

   cdlPrinter.ShowPrinter

   Printer.Copies = cdlPrinter.Copies

   Printer.Orientation = cdlPrinter.Orientation

   Printer.EndDoc

## Skill Practice;

- Observe the windows form and fill out the format

| Control Name | Control<br>Property Name | Control<br>Property Value | Event Name |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Part A: Ex No: 6 - Store Details of students using ADO.Net

## Aim:

Develop a database application to store the details of students using ADO.NET

## Pre-Learning Skills

- Understanding the SQLServer database creation
- Known about Table creation, Column creation with data type and constraints
- User Interaction forms

## Procedure:

1. Open a New project.
2. Choose the **windows application** from the Templates.
3. Open Server explorer from the menu
4. Right click the data connection and choose Add New connection.
5. Select the database file and path and click create new one
6. Expand the Database and select tables
7. Right Click and select add new table
8. Design the table as per the students data
9. Click Update to save the table
10. Design the Form as follows

➢ Change the Form Caption to the Title of the program

➢ Drag and Drop required Controls from the tool box and change the name and text properties.

➢ Place a Save button and Close button.

11. Write the following code in the click () event of all the buttons.
    1. Open a connection using SQL Provider.
    2. Execute Query using Command
    3. Close the connection
12. Save and Execute (press F5).

## Source Code:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
namespace Ex_6
{
    public partial class Form1 : Form
    {
       public Form1()
        {
            InitializeComponent();
        }
        // ADO.Net Connection Object
        SqlConnection con = new SqlConnection();
        //ADO.Net Command Object
        SqlCommand cmd = new SqlCommand();
        string sql;
        private void btn_save_Click(object sender, EventArgs e)
        {
            //Connection Property
            con.ConnectionString = @"Data
            Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" +
            Application.StartupPath + @"\student_db.mdf;Integrated
            Security=True";
          System.Diagnostics.Debug.WriteLine(con.ConnectionString);
            // Open Connection
            con.Open();
          //Parameterized SQL Command for INSERT Records in a TABLE
            sql = "insert into
           student_master(rollno,student_name,dept_name,address,
           mobileno)values(@rno,@stuname,@dept,@address,@mobile)”;
            cmd = new SqlCommand(sql, con);
            // Add Parameter Value
           cmd.Parameters.Add("@rno",SqlDbType.Int).Value =
           txt_rno.Text;
           cmd.Parameters.Add("@stuname",SqlDbType.VarChar).Value=
              txt_stuname.Text;
           cmd.Parameters.Add("@dept", SqlDbType.VarChar).Value =
           cmb_deptname.SelectedItem.ToString();
           cmd.Parameters.Add("@address",SqlDbType.VarChar).Value=
           txt_address.Text;
           cmd.Parameters.Add("@mobileno",SqlDbType.VarChar).Value=
           txt_mobile.Text;
           //Execute method return integer value
           int result = cmd.ExecuteNonQuery();
               if (result == 1)
                 MessageBox.Show("Student's Data Inserted.");
             else
                MessageBox.Show("Student's Data Not Inserted.");
            // Close Connection
         con.Close();
        }
}`      }
```

**Output:** (Print your screen for output)

**Result:**(Write down the result)

## Viva Questions

1. **What is ADO.NET**?

ADO.NET is the primary relational data access model for Microsoft .NET-based applications. It may be used to access data sources.

ADO.NET provides mainly the following two types of architectures:

- Connected Architecture.
- Disconnected Architecture.

2. **Define different execute methods of ADO.NET command object?**

- **ExecuteScalar: -** This method returns a single value from the first row and first column of the result get from the execution of SQL query.
- **ExecuteNonQuery**: - This method executes the DML SQL query just like insert, delete or update and then returns the number of rows affected by the action.

3. **What are the different namespaces available in ADO.NET?**

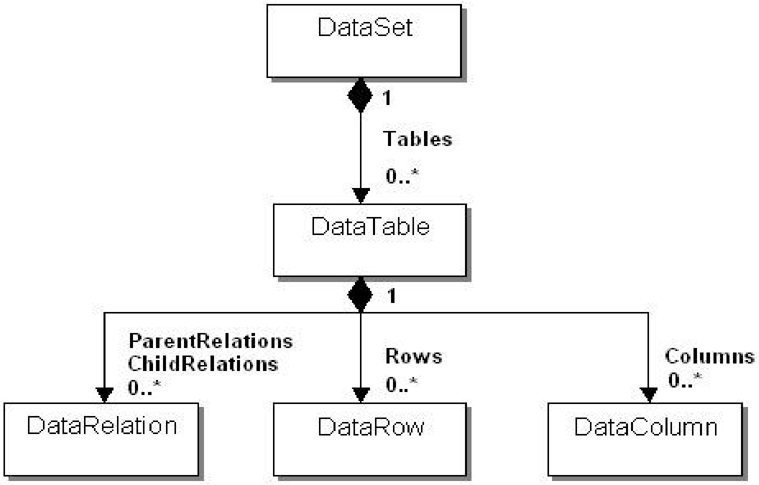
**System.Data:** It contains the definition for rows, columns, relations, views, tables, constraints, and databases.

**System.Data.SqlClient:** It is a collection of classes that are helpful in connecting to a Microsoft SQL Server 2008 database such as SqlConnection, SqlCommand, SqlDataAdapter, etc.

**System.Data.Odbc:** It consists of classes that are required for connecting with most Odbc Drivers. These classes include OdbcConnection, OdbcCommand.

**System.Data.OracleClient:** It has classes required for connection with an Oracle database, OracleConnection, OracleCommand

4. **How can you define the DataSet structure?**



5. **What are the different namespaces available in ADO.NET?**

**System.Data:** It contains the definition for rows, columns, relations, views, tables, constraints, and databases.

**System.Data.SqlClient:** It is a collection of classes that are helpful in connecting to a Microsoft SQL Server 2008 database such as SqlConnection, SqlCommand, SqlDataAdapter, etc.

**System.Data.Odbc:** It consists of classes that are required for connecting with most Odbc Drivers. These classes include OdbcConnection, OdbcCommand.

**System.Data.OracleClient:** It has classes required for connection with an Oracle database, OracleConnection, OracleCommand

## Skill Practice;

❖ Observe the windows form and fill out the format

| Control Name | Control Property Name | Control Property Value |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

- Observe the ADO.Net Code and fill out the format

| Command Object | Properties used | Method used if any |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

- Design a user interactive form for employee management

# Part A: Ex No: 7 - Create an ASP.Net webpage for login entry

## Aim:

Create a simple ASP.NET page to Output Text with a form, two HTML text boxes, an HTML button, and an HTML <span> element. Create an event procedure for the button.

## Procedure:

1. Open Visual Studio –> Create New Empty Website
2. Add New Web forms
3. Design Login page with Two TextBox along with Button control as below

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

<!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>Login Page</title>
</head>
<body>
  <form id="form1" runat="server">
  <div align="center">
  <fieldset style ="width:200px;">
  <legend>Login page </legend>
    <span id=”user” style="color: red;">
           <asp:label ID =”label1” runat=”server”>Enter a <span>User
       Name</span>
    </span>
     <asp:TextBox ID="txtusername" placeholder="username"
runat="server"
        Width="180px"></asp:TextBox>
     <br />
     <br />
    <span id=”pass” style="color: red;">
           <asp:label ID =”label1” runat=”server”>Enter a
       <span>Password</span>
    </span>

     <asp:TextBox ID="txtpassword" placeholder="password"
runat="server"
        Width="180px" TextMode="Password"></asp:TextBox>
     <br />
```

```
            <br />
            <asp:Button ID="btnsubmit" runat="server" Text="Submit"
              Width="81px" onclick="btnsubmit_Click" />
              <br />
             <span id=”user” style="color: red;">
                  <asp:label ID =”lbl_result” runat=”server”>Enter a <span>User
               Name</span>
           </span>

        </fieldset>
        </div>

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

4. Write C# code on Login Button click events.

**Source Code:**

```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Ex_7
{
    public partial class login : System.Web.UI.Page
    {
        protected void btn_login_Click(object sender, EventArgs e)
        {
            if (txtusername.Text == "")
                lbl_result.Text = "Enter UserName";
            else if (txtpassword.Text == "")
                lbl_result.Text = "Enter Password";
            else
            {
           // check whether given username and password is equal to
           //   “Computer” & “mspvl”
                if (txtusername.Text == "Computer" &&
                              txtpassword.Text == "mspvl")
                    lbl_result.Text = "Login Success";
                else
                    lbl_result.Text = "Invalid Name or Password";
            }
        }
    }
}

```

**Output:** (Print your screen for output)

**Result:**(Write down the result)

**Viva Questions;**

1. **What does <span> HTML Tag do?**

   The <span> element is the inline equivalent to the block-level <div> element.

   It is used to select inline content for purely stylistic purposes.

2. **What is static and dynamic login form?**

   Static Login Form

   Username and password are fixed
   Predefined user can login
   Does not use database
   Write username and password values on coding page

   Dynamic Login Form

   Username and Password are not fixed
   Multiple use can login
   Use Database
   Fetch username and password values from database