

PRACTICAL 1: Working with Basic C# and ASP.NET.

a. Create an application that obtain four int values from the user and display the product

```
using System;
namespace practical1
{
    internal class program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Faizan");
            int a, b, c, d, ans, ans2, ans3;
            Console.WriteLine("Enter no1: ");
            a = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter no2: ");
            b = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter no3: ");
            c = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter no4: ");
            d = int.Parse(Console.ReadLine());
            ans = a * b * c * d;
            Console.WriteLine("Product: " + ans);
            ans2 = b + c + d;
            Console.WriteLine("sum: " + ans2);
            ans3 = a / b;
            Console.WriteLine("Divide" + ans3);
            Console.ReadKey();
        }
    }
}
```

b. Create an application to demonstrate string operations

```
//string operation
using System;

namespace practical2
{
    internal class program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Faizan");
            string str = "Chetana's College";
            string str1 = "Chetana's institute";
            Console.WriteLine(str);
            Console.WriteLine(str1);
            int length = str.Length;
            Console.WriteLine("Length: " + length);
            string letter = str.ToUpper();
            Console.WriteLine("Capital: " + letter);
            string word = str.ToLower();
            Console.WriteLine("Small: " + word);
            int A = str.LastIndexOf('e');
            Console.WriteLine("Last Index Number: " + A);
            string name = str + str1;
            Console.WriteLine("Concatation: " + name);
            int B = str.CompareTo(str1);
            Console.WriteLine("Comparison of strings: " + B);
            Console.ReadKey();
        }
    }
}
```

PRACTICAL 2: Working with Object Oriented C# and ASP.NET.

a. Create an application that receive the (student l'd, student name, course name, date of birth) information from a set of students. The application should also display the information of all the students once the data entered.

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

namespace practical3
{
    class Program
    {
        static void Main(string[] args)
        {
            string name, course, DOB;
            int rollno;
            Console.WriteLine("Mohd Faizan");
            Console.WriteLine("enter students details");
            for (int i = 0; i <= 2; i++)
            {
                Console.Write("Roll No : ");
                rollno = int.Parse(Console.ReadLine());
                Console.Write("name : ");
                name = (Console.ReadLine());
                Console.Write("course : ");
                course = (Console.ReadLine());
                Console.Write("DOB : ");
                DOB = (Console.ReadLine());

                Console.WriteLine("Roll No : " + rollno);
                Console.WriteLine("name : " + name);
                Console.WriteLine("course : " + course);
                Console.WriteLine("date of birth : " + DOB);
            }
            Console.ReadKey();
            Console.ReadLine();
        }
    }
}
```

PRACTICAL 3: Create an application to demonstrate following operations:

a. Generate Fibonacci series

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

public class FibonacciExample
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Mohd Faizan");
        int n1 = 0, n2 = 1, n3, i, number;
        Console.Write("Enter the number of elements: ");
        number = int.Parse(Console.ReadLine());
        Console.Write(n1 + " " + n2 + " "); //printing 0 and 1
```

```

        for (i = 2; i < number; ++i) //loop starts from 2 because 0 and 1 are
already printed
        {
            n3 = n1 + n2;
            Console.Write(n3 + " ");
            n1 = n2;
            n2 = n3;
        }
        Console.ReadLine();
    }
}

```

b. Test for primary numbers

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

public class PrimeNumberExample
{
    public static void Main(string[] args)
    {
        int n, i, m = 0, flag = 0;
        Console.Write("Mohd Faizan");
        Console.Write("\n Enter the Number to check Prime: ");
        n = int.Parse(Console.ReadLine());
        m = n / 2;
        for (i = 2; i <= m; i++)
        {
            if (n % i == 0)
            {
                Console.Write("Number is not Prime.");
                flag = 1;
                break;
            }
        }
        if (flag == 0)
            Console.Write("Number is Prime.");
        Console.ReadLine();
    }
}

```

c. Test for vowels

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

public class exercise16
{
    static void Main(string[] args)
    {
        char ch;
        Console.Write("Mohd Faizan ");
        Console.Write("\n\n");
        Console.Write("check whether the input alphabet is a vowel or not:\n");
        Console.Write("-----");
        Console.Write("\n\n");
    }
}

```

```

Console.Write("Input an Alphabet (A-Z or a-z) : ");
ch = Convert.ToChar(Console.ReadLine().ToLower());
int i = ch;
if (i >= 48 && i <= 57)
{
    Console.Write("You entered a number, Please enter an alphabets.");
}
else
{
    switch (ch)
    {
        case 'a':
            Console.WriteLine("The Alphabet is vowel");
            break;
        case 'i':
            Console.WriteLine("The Alphabet is vowel");
            break;
        case 'o':
            Console.WriteLine("The Alphabet is vowel");
            break;
        case 'u':
            Console.WriteLine("The Alphabet is vowel");
            break;
        case 'e':
            Console.WriteLine("The Alphabet is vowel");
            break;
        default:
            Console.WriteLine("The Alphabet is not a vowel");
            break;
    }
    Console.ReadKey();
}
}

```

d. Use of foreach loop with arrays

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class Practical7
{
    public static void Main()
    {
        Console.WriteLine("Mohd Faizan");
        String[] arr = { "Yashraaj", "Faizan", "Abhinash", "iqbal" };

        Console.WriteLine();

        Console.WriteLine("Array printing using foreach loop = ");

        foreach (String ch in arr)
        {
            Console.WriteLine(ch);
        }
        Console.ReadLine();
    }
}

```

e. Reverse a number and find sum of digits of a number.

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

namespace ConsoleApp2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Faizan");
            int n, r, sum = 0, temp, iqbal = 0;
            Console.Write("Enter the Number: ");
            n = int.Parse(Console.ReadLine());
            temp = n;
            while (n > 0)
            {
                r = n % 10;
                iqbal += r;
                sum = (sum * 10) + r;

                n = n / 10;
            }

            Console.WriteLine(sum);
            Console.Write(iqbal);
            Console.ReadKey();
        }
    }
}
```

PRACTICAL 4: Create simple application to perform following operations

a. Finding factorial value

```
using System;
public class FactorialExample
{
    public static void Main(string[] args)
    {
        Console.WriteLine("mohd iqbal");
        int i, fact = 1, number;
        Console.Write("Enter any Number: ");
        number = int.Parse(Console.ReadLine());
        for (i = 1; i <= number; i++)
        {
            fact = fact * i;
        }
        Console.WriteLine("Factorial of " + number + " is: " + fact);
        Console.ReadLine();
    }
}
```

b. Money conversion

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

namespace Money
```

```

{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Iqbal");
            double usd, inr, val;
            Console.WriteLine("Enter the USD: ");
            usd = int.Parse(Console.ReadLine());
            val = 82.69;
            inr = usd * val;
            Console.WriteLine("{0} Dollar = {1} INR ", usd, inr);
            Console.ReadLine();
        }
    }
}

c. Temperature conversion
using System;

namespace TemperatureConverter
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Iqbal");
            Console.WriteLine("Temperature Converter");
            Console.WriteLine("-----");

            while (true)
            {
                Console.WriteLine("Enter 1 to convert from Celsius to Fahrenheit");
                Console.WriteLine("Enter 2 to convert from Fahrenheit to Celsius");
                Console.WriteLine("Enter 3 to exit");
                Console.Write("Choice: ");

                int choice;
                if (!int.TryParse(Console.ReadLine(), out choice))
                {
                    Console.WriteLine("Invalid choice. Please enter a valid option.");
                    continue;
                }

                if (choice == 1)
                {
                    Console.Write("Enter temperature in Celsius: ");
                    if (double.TryParse(Console.ReadLine(), out double celsius))
                    {
                        double fahrenheit = (celsius * 9 / 5) + 32;
                        Console.WriteLine($"Temperature in Fahrenheit: {fahrenheit:F2}");
                    }
                    else
                    {
                        Console.WriteLine("Invalid input. Please enter a valid number.");
                    }
                }
                else if (choice == 2)

```

```

        {
            Console.WriteLine("Enter temperature in Fahrenheit: ");
            if (double.TryParse(Console.ReadLine(), out double
fahrenheit))
            {
                double celsius = (fahrenheit - 32) * 5 / 9;
                Console.WriteLine($"Temperature in Celsius:
{celsius:F2}");
            }
            else
            {
                Console.WriteLine("Invalid input. Please enter a valid
number.");
            }
        }
        else if (choice == 3)
        {
            Console.WriteLine("Exiting the application...");
            break;
        }
        else
        {
            Console.WriteLine("Invalid choice. Please enter a valid
option.");
        }
    }
}
}
}

```

PRACTICAL 5: Create simple application to demonstrate use of following concepts:

a. Function overloading

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Function_Overloading
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Iqbal");
            Calculator.sum();
            Calculator.sum(5, 4);
            Calculator.sum(9.3f, 8.6f);
            Calculator.sum("Hello World");
            Console.Read();
        }
    }
    static class Calculator
    {
        public static void sum()
        {
            Console.WriteLine("No Value Provided");
        }
        public static void sum(int x, int y)
        {
            Console.WriteLine("Sum of {0} and {1} is {2}", x, y, (x + y));
        }
    }
}

```

```

        public static void sum(float x, float y)
        {
            Console.WriteLine("Sum of {0} and {1} is {2}", x, y, (x + y));
        }
        public static void sum(string s)
        {
            Console.WriteLine("{0} - is not a numeric value", s);
        }
    }
}

b. Inheritance [all types]
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

// single inheritance
class Animal
{
    public void Eat()
    {
        Console.WriteLine("Yasraaj Kashid");
        Console.WriteLine("Animal is eating.");
    }
}

class Dog : Animal
{
    public void Bark()
    {
        Console.WriteLine("Dog is barking.");
    }
}

// multi-level inheritance
class Mammal : Animal
{
    public void Run()
    {
        Console.WriteLine("Mammal is running.");
    }
}

class Horse : Mammal
{
    public void Gallop()
    {
        Console.WriteLine("Horse is galloping.");
    }
}

// hierarchical inheritance
class Bird : Animal
{
    public void Fly()
    {
        Console.WriteLine("Bird is flying.");
    }
}

class Eagle : Bird
{
    public void Hunt()
    {
        Console.WriteLine("Eagle is hunting.");
    }
}

```



```

    }
}
class Penguin : Bird
{
    public void Swim()
    {
        Console.WriteLine("Penguin is swimming.");
    }
}
// multiple inheritance
interface I1
{
    void Method1();
}
interface I2
{
    void Method2();
}
class MyClass : I1, I2
{
    public void Method1()
    {
        Console.WriteLine("Method1 is called.");
    }
    public void Method2()
    {
        Console.WriteLine("Method2 is called.");
    }
}

// main program
class Program
{
    static void Main(string[] args)
    {
        // single inheritance
        Dog dog = new Dog();
        dog.Eat();
        dog.Bark();

        // multi-level inheritance
        Horse horse = new Horse();
        horse.Eat();
        horse.Run();
        horse.Gallop();

        // hierarchical inheritance
        Eagle eagle = new Eagle();
        Penguin penguin = new Penguin();
        eagle.Fly();
        eagle.Hunt();
        penguin.Fly();
        penguin.Swim();

        // multiple inheritance
        MyClass myClass = new MyClass();
        myClass.Method1();
        myClass.Method2();

        Console.ReadLine();
    }
}

```

c. Constructor overloading

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

namespace Constructor_Overloading
{
    class GameScore
    {
        string user;
        int age;
        //Default Constructor
        public GameScore()
        {
            user = "Steven";
            age = 28;
            Console.WriteLine("Previous User {0} and he was {1} year old", user,
age);
        }

        //Parameterized Constructor
        public GameScore(string name, int age1)
        {
            user = name;
            age = age1;
            Console.WriteLine("Current User {0} and he is {1} year old", user,
age);
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mohd Iqbal");
            GameScore gs = new GameScore(); //Default Constructor Called
            GameScore gs1 = new GameScore("Clark", 35); //Overloaded Constructor.
            Console.ReadLine();
        }
    }
}
```

PRACTICAL 6: Working with Database.

A. Create a simple web page with various server controls to demonstrate setting and use of their properties [Example : AutoPostBack]

Web.aspx.cs

```
using System;

namespace WebApplication1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Initialize controls or perform one-time setup here
            }
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

    {
        // Handle button click event
        string name = txtName.Text;
        lblMessage.Text = "Hello, " + name + "! Submission successful.";
    }

    protected void txtAutoPostBack_TextChanged(object sender, EventArgs e)
    {
        // Handle AutoPostBack for TextBox
        lblAutoPostBackResult.Text = "Text changed: " + txtAutoPostBack.Text;
    }
}

```

Web.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Server Controls Demo</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Server Controls Demo</h2>

        <!-- TextBox Control -->
        <asp:TextBox ID="txtName" runat="server" placeholder="Enter your
name"></asp:TextBox>
        <br />

        <!-- Button Control -->
        <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />
        <br />

        <!-- Label Control -->
        <asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>
        <br />

        <!-- DropDownList Control -->
        <asp:DropDownList ID="ddlColors" runat="server">
            <asp:ListItem Text="Red" Value="Red"></asp:ListItem>
            <asp:ListItem Text="Green" Value="Green"></asp:ListItem>
            <asp:ListItem Text="Blue" Value="Blue"></asp:ListItem>
        </asp:DropDownList>
        <br />

        <!-- CheckBox Control -->
        <asp:CheckBox ID="chkAgree" runat="server" Text="I agree to the terms and
conditions" />
        <br />

        <!-- AutoPostBack example with TextBox -->
        <asp:TextBox ID="txtAutoPostBack" runat="server" AutoPostBack="true"
OnTextChanged="txtAutoPostBack_TextChanged"></asp:TextBox>
        <br />

        <!-- Display AutoPostBack result -->
        <asp:Label ID="lblAutoPostBackResult" runat="server" Text=""></asp:Label>
    </form>
</body>
</html>

```

```

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

```

B. Demonstrate the use of calendar control to perform following operations:

Display messages in a calendar control

Display vacation in a calendar control

Selected day in a calendar control using style

Difference between two calendar dates

Calender.aspx.cs

```

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

namespace practical6B.vacation
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Calendar1_SelectionChanged(object sender, EventArgs e)
        {
            Calendar1.SelectedDayStyle.BackColor = System.Drawing.Color.Yellow;
            Calendar1.SelectedDayStyle.BorderColor = System.Drawing.Color.Red;
            Calendar1.SelectedDayStyle.ForeColor = System.Drawing.Color.Red;
        }

        protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
        {
            if (e.Day.Date.Year == 2023 && e.Day.Date.Month == 12 &&
e.Day.Date.Day == 31)
            {
                Label l1 = new Label();
                l1.Text = "<br>New Year";
                e.Cell.BackColor = System.Drawing.Color.Red;
                e.Cell.Controls.Add(l1);
            }

            if (e.Day.Date.Year == 2023 && e.Day.Date.Month == 12 &&
e.Day.Date.Day == 16)
            {
                Label l1 = new Label();
                l1.Text = "<br>Annual Day!";
                //e.Cell.BackColor = System.Drawing.Color.Red;
                e.Cell.Controls.Add(l1);
            }
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            TimeSpan ts = Calendar1.SelectedDate - Calendar2.SelectedDate;
            Label1.Text = ts.TotalDays.ToString();
        }
    }
}

```

PRACTICAL 7: Create a registration form to demonstrate use of various validation controls

Validation.aspx.cs

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

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

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            if (Page.IsValid) return;
            lblresult.Text = "Registration is Successful...";
        }

        protected void CustomValidator1_ServerValidate(object source,
ServerValidateEventArgs args)
        {
            int x = int.Parse(args.Value);
            if (x % 2 == 0)
                args.IsValid = true;
            else
                args.IsValid = false;
        }

        protected void TextBox1_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

PRACTICAL 8: create a web application bind data in a multiline textbox by querying in another textbox.

Bind.aspx.cs

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

namespace databind_8
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        public String txtdata;
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```

        protected void TextBox2_TextChanged(object sender, EventArgs e)
        {
            textdata = TextBox2.Text;
            this.DataBind();
        }
    }
}

Bind.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="databind_8.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox1" Text='<%# textdata %>' runat="server"
Height="124px" TextMode="MultiLine" Width="210px"></asp:TextBox>
            <asp:TextBox ID="TextBox2" runat="server" AutoPostBack="True"
OnTextChanged="TextBox2_TextChanged"></asp:TextBox>
        </div>
    </form>
</body>
</html>

```

PRACTICAL 9: Create a web application to display data binding using dropdown list control

Web.aspx.cs

```

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

namespace databind_dd
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            DataSet studs = new DataSet();
            studs.Tables.Add(new DataTable("Students"));
            studs.Tables["Students"].Columns.Add(new DataColumn("Roll no"));
            studs.Tables["Students"].Columns.Add(new DataColumn("Name"));
            studs.Tables["Students"].Columns.Add(new DataColumn("Class"));
            studs.Tables["Students"].Columns.Add(new DataColumn("Phone no"));
            studs.Tables["Students"].Columns.Add(new DataColumn("Email"));

            DataRow dr = studs.Tables["Students"].NewRow();
            dr["Roll no"] = "11";
            dr[1] = "Sahil";
            dr[2] = "IT";
            dr[3] = "12345687";
            dr[4] = "sahil@g.com";
            studs.Tables["Students"].Rows.Add(dr);
        }
    }
}

```

```

        dr = studs.Tables["Students"].NewRow();
        dr["Roll no"] = "12";
        dr[1] = "Pawan";
        dr[2] = "IT";
        dr[3] = "1234568790";
        dr[4] = "pawan@g.com";
        studs.Tables["Students"].Rows.Add(dr);

        GridView1.DataSource = studs;
        GridView1.DataBind();

        DropDownList1.DataSource = studs;
        DropDownList1.DataTextField = "Name";
        DropDownList1.DataValueField = "Roll no";
        DropDownList1.DataBind();
    }
}

```

Web.aspx.cs

```

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

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server">
            </asp:GridView>
            <br />
            <br />
            <asp:DropDownList ID="DropDownList1" runat="server">
            </asp:DropDownList>
        </div>
    </form>
</body>
</html>

```