1)Write a program to demonstrate delegates in C#.

Introduction:

C# delegates are similar to pointers to functions, in C or C++. A delegate is a reference type variable that holds the reference to a method. The reference can be changed at runtime.

Delegates are especially used for implementing events and the call-back methods. All delegates are implicitly derived from the System.Delegate class.

Delegate declaration determines the methods that can be referenced by the delegate. A delegate can refer to a method, which has the same signature as that of the delegate.

Once a delegate type is declared, a delegate object must be created with the new keyword and be associated with a particular method. When creating a delegate, the argument passed to the new expression is written similar to a method call, but without the arguments to the method.

Syntax:

```
delegate <return type> <delegate-name> <parameter list>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication4
{
    class PaperPrinter
        public void Print(string message)
        {
            Console.WriteLine(message);
        }
    }
    class ScreenPrinter
        public void Print(string message)
            Console.WriteLine(message);
    }
```

```
public delegate void PrintDelegate(string message);
    class Program
        static void Main(string[] args)
        {
            PrintDelegate pdObject;
            PaperPrinter pp = new PaperPrinter();
            pdObject = pp.Print;
            pdObject("hello paper printer");
            Console.ReadLine();
            PrintDelegate sdObject;
            ScreenPrinter sc = new ScreenPrinter();
            sdObject = sc.Print;
            sdObject("hello screen printer");
            Console.ReadLine();
        }
    }
}
```

```
C:\Windows\system32\cmd.exe
```

hello screen printer

Press any key to continue . . . _

2) Write a LINQ program to select Student whose address is Kathmandu and Mark is greater than 2.5 Introduction:

LINQ (Language Integrated Query) is uniform query syntax in C# and VB.NET to retrieve data from different sources and formats. It is integrated in C# or VB, thereby eliminating the mismatch between programming languages and databases, as well as providing a single querying interface for different types of data sources. For example, SQL is a Structured Query Language used to save and retrieve data from a database. In the same way, LINQ is a structured query syntax built in C# and VB.NET to retrieve data from different types of data sources such as collections, ADO.Net DataSet, XML Docs, web service and MS SQL Server and other databases.

Syntax:

```
from <range variable> in <IEnumerable<T> or IQueryable<T> Collection>
      <Standard Query Operators> <lambda expression>
      <select or groupBy operator> <result formation>
Code:
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace LINQ{
    public class Student{
        public int id { get; set; }
        public string name { get; set; }
        public int marks{ get; set; }
        public string address{ get; set; }
        public static <Student> GetAllStudents(){
             List<Student> listStudents = new List<Student>(){
```

```
new Student {ID=101, name ="Ashish", marks=2.4, address="Pokhara"},
         new Student {ID=102, name="Rabin", marks=3.1, address ="Kathmandu"},
         new Student {ID=103, name="Nakul", marks =2.2, address ="Kusma"},
         new Student {ID=104, name="Muna",marks =1.7,address ="Kathmandu "},
    };
            return listStudents;
        }
    }
    class Program{
        static void Main(string[] args){
            var selected_std= Student.GetAllStudents(.
            Where(std => std.address == "Kathmandu").
            Max(std => std.marks);
            Console.WriteLine("Student whose address is kathmandu and
            Mark is greater than 2.5 =" +selected_std);
            Console.ReadKey();
        }
    }
}
```

Select C:\Windows\system32\cmd.exe

Student whose address is Kathmandu and Mark is greater than 2.5 =Rabin Press any ley to continue . . . $_$

3) Write a program to create user login form in ASP.NET web page and display username in welcome page with a logout link.

Introduction:

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.

ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

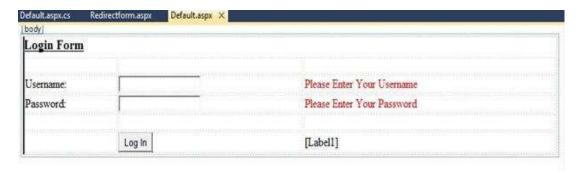
ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework. The ASP.NET application codes can be written in any of the following languages: C#, Visual Basic.Net, Jscript, J#

Code:

Login.aspx

```
>
         Username:
         >
           <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
         >
           <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"</pre>
ControlToValidate="TextBox1" ErrorMessage="Please Enter Your Username"
ForeColor="Red"></asp:RequiredFieldValidator>
         >
         word:
         <asp:TextBox ID="TextBox2" TextMode="word" runat="server"></asp:TextBox>
         >
           <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox2" ErrorMessage="Please Enter Your word"
```

```
ForeColor="Red"></asp:RequiredFieldValidator>
       >
       >
>
       >
        <asp:Button ID="Button1" runat="server" Text="Log In" onclick="Button1 Click" />
       >
        <asp:Label ID="Label1" runat="server"></asp:Label>
       </div>
 </form>
```



Login.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;
using System.Data.SqlClient;
public partial class Default: System.Web.UI.Page
  protected void Page_Load(object sender, EventArgs e)
  protected void Button1_Click(object sender, EventArgs e)
    SqlConnection con = new SqlConnection(@
```

```
"Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Database.mdf;Integrated
Security=True;User Instance=True");
    SqlCommand cmd = new SqlCommand("select * from tbl data where username=@username and
word=@word", con);
    cmd.Parameters.AddWithValue("@username", TextBox1.Text);
    cmd.Parameters.AddWithValue("word", TextBox2.Text);
    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    sda.Fill(dt);
    con.Open();
    int i = cmd.ExecuteNonQuery();
    con.Close();
    if (dt.Rows.Count > 0)
{
      Session["id"] = TextBox1.Text;
      Response.Redirect("Redirectform.aspx");
      Session.RemoveAll();
    } else
{
      Label1.Text = "You're username and word is incorrect";
      Label1.ForeColor = System.Drawing.Color.Red;
```



```
RedirectForm.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Redirectform.aspx.cs"
Inherits="Redirectform" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</pre>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html
  xmlns="http://www.w3.org/1999/xhtml">
  <head runat="server">
    <title></title>
  </head>
  <body bgcolor="#ffff99">
    <form id="form1" runat="server">
      <div>
         >
           <strong style="font-size: xx-large">Hello Everyone! Welcome to my Page.
</strong>
```

```
</div>
      <asp:Image ID="Image1" runat="server" Height="335px"
ImageUrl="~/2.jpg" Width="817px" />
      >
>
        <asp:Label ID="Label1" runat="server"></asp:Label>
      >
        <asp:Button ID="Button1" runat="server" Height="47px" onclick="Button1 Click"
Text="Logout" Width="92px" />
      </form>
  </body>
</html>
```

RedirectForm.aspx.cs

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class Redirectform: System.Web.UI.Page
```

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

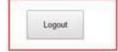
{
    Label1.Text = Session["id"].ToString();
}
protected void Button1_Click(object sender, EventArgs e)

{
    Session.RemoveAll();
    Response.Redirect("Default.aspx");
}
```

Hello Everyone! Welcome to my Page.



Ashish Poudel



4) Write a C# program to show insert and select operation in database from Windows Form / Desktop Application.

Introduction:

Windows Forms (WinForms) is a free and open-source graphical (GUI) class library included as a part of Microsoft .NET, .NET Framework or Mono Framework,[1] providing a platform to write client applications for desktop, laptop, and tablet PCs. All visual elements in the Windows Forms class library derive from the Control class. This provides the minimal functionality of a user interface element such as location, size, color, font, text, as well as common events like click and drag/drop.

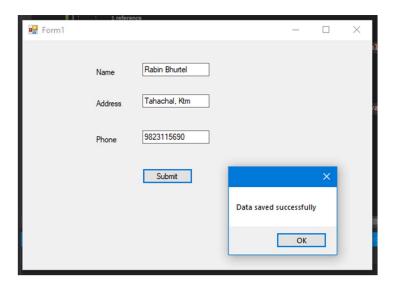
Accessing Data from a database is one of the important aspects of any programming language. It is an absolute necessity for any programming language to have the ability to work with databases. C# is no different. It also can work with new forms of databases such as MongoDB and MySQL. C# and .Net can work with a majority of databases, the most common being Oracle and Microsoft SQL Server. But with every database, the logic behind working with all of them is mostly the same.

```
using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
namespace WindowsFormsApp1{
    public partial class Form1 : Form{
        public Form1(){
            InitializeComponent();
        private void label1 Click(object sender, EventArgs e){
        }
        private void label2 Click(object sender, EventArgs e){
        }
        private void label3_Click(object sender, EventArgs e){
        }
        private void button1 Click(object sender, EventArgs e){
            MySqlConnection conn = new MySqlConnection("server = localhost;
database = studentsdb; uid = root; pwd = ");
```

```
MySqlCommand cmd = null;
    string cmdString = "";
    conn.Open();

cmdString = "Insert into Students(name, address, phone) values ('
" + textBox1.Text + " ', ' " + textBox2.Text + " ', ' " + textBox3.Text + "
')";

cmd = new MySqlCommand(cmdString, conn);
    cmd.ExecuteNonQuery();
    conn.Close();
    MessageBox.Show("Data saved successfully");
    }
}
```



In Database:



5) Write a program to demonstrate different types of lambda expression used in C#.

Introduction:

Lambda expressions in C# are used like anonymous functions, with the difference that in Lambda expressions you don't need to specify the type of the value that you input thus making it more flexible to use. The '=>' is the lambda operator which is used in all lambda expressions. The Lambda expression is divided into two parts, the left side is the input and the right is the expression.

The Lambda Expressions can be of two types:

Expression Lambda: Consists of the input and the expression.

Statement Lambda: Consists of the input and a set of statements to be executed.

Syntax:

```
input => expression; or
input => { statements };
```

```
Using Expressions lambda:
1
2
Using Statements lambda:
1
2
Press any key to continue . . . .
```

6) Write a LINQ Program to find the highest salary of Sales department.

Introduction:

The official goal of the LINQ family of technologies is to add "general purpose query facilities to the .NET Framework that apply to all sources of information, not just relational or XML data".

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace LINQ{
   public class Employee{
        public int ID { get; set; }
        public string Name { get; set; }
        public int salary { get; set; }
        public string Department { get; set; }
        public static List<Employee> GetAllEmployees(){
            List<Employee> listEmpolyees = new List<Employee>(){
        new Employee {ID=101,Name="jk",salary=20,Department="sales"},
         new Employee {ID=101,Name="jk",salary=30,Department="sales"},
         new Employee {ID=101,Name="jk",salary=400,Department="sales"},
         new Employee {ID=101,Name="jk",salary=240,Department="sales"},
    };
            return listEmpolyees;
        }
```

```
}
    class Program{
        static void Main(string[] args){
            var mshighestSalary = Employee.GetAllEmployees().Where(emp =>
emp.Department == "sales")
                .Max(emp => emp.salary);
            Console.WriteLine("sales depatrment highest salary =" +
mshighestSalary);
            Console.ReadKey();
        }
    }
}
```

```
Select C:\Windows\system32\cmd.exe
sales depatrment highest salary =400
Press any key to continue . . . _
```