LAB - 1

Write a win form application to show name of 5 different university maintained on a list to a message box.

~~using~~ Source code :

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

    namespace UniversityListApp
        {
    public partial class MainForm : Form
        {
    private List <string> universities = new List <string>
        {
      "Tribhuvan University",
      "Pokhara University",
      "Kathmandu University",
      "Purbanchal University",
      "Nepal Sanskrit University"
            };
    public MainForm ( )
        {
      InitializeComponent ( );
        }
private void showButton_Click (object, sender, EventArgs e)
        {
  string universitiesList = string.Join (Environment.NewLine,
                                         universities);
  MessageBox.Show (universitiesList, "List of Universities");
        }
    }
```

LAB - 2

Write a program to display the use of case when statement to display area of different types of shape.

A program to display the use of case when statement to display area of different types of shape a is:

Source code:

```
using System;
    class Program
        {
    static void Main (string [ ] arge)
        {
    Console. WriteLine ("Choose a shape to calculate its area : ");
    Console. WriteLine ("1. Rectangle");
    Console. WriteLine ("2. Circle");
    Console. WriteLine ("3. Triangle");
    Console. WriteLine ("4. Square");
    Console. Write Line ("5. Exit");
int choice = Convert. ToInt 32 (Console. ReadLine ( ));
        switch (choice)
        {
    case 1:
Calculate Rectangle Area ( );
    break;
    case 2:
Calculate Circle Area ( );
    break;
    case 3:
Calculate Triangle Area ( );
    break;
```

```
    case 4:
Calculate Square Area ( );
    break;
    case 5:
Console. WriteLine ("Exiting ....");
    break;
default:
    Console. WriteLine ("Invalid choice!");
        break;
        }
        }
Static void Calculate Rectangle Area ( )
        {
    Console. WriteLine ("Enter length of the rectangle :");
    double length = Convert. To Double (Console. Read Line ( ));
    Console. WriteLine ("Enter width of the rectangle :");
    double width = Convert. To Double (Console. Read Line ( ));
            double area = length * width;
    Console. WriteLine ($"Area of the rectangle : {area}");
                }
    Static void Calculate Circle Area ( )
        {
    Console. WriteLine ("Enter radius of the circle :");
    double radius = Convert. To Double (Console. ReadLine ( ));
        double area = Math. PI * radius * radius;
    Console. WriteLine ($"Area of the circle : {area}");
                }
    static void Calculate Triangle Area ( )
            {
    Console. WriteLine ("Enter base of the triangle :");
double @base = Convert. toDouble (Console. Read line ( ));
```

```
Console.WriteLine("Enter height of the triangle:");
double @base = Convert.ToDouble(Console.ReadLine());
    double area = 0.5 * @base * height;
Console.WriteLine($"Area of the triangle: {area}");
        }
  static void CalculateSquareArea()
        {
Console.WriteLine("Enter side length of the square:");
  double side = Convert.ToDouble(Console.ReadLine());
    double area = side * side;
Console.WriteLine($"Area of the square: {area}");
        }
    }
```

LAB-3

Write a program to create a custom exception class and handle it using different level of try catch statement.

A program to create a custom exception class and handle it using different level of try catch statement is:

Source code:

```
using System;
public class CustomException : Exception
    {
public CustomException (string message) : base (message)
        {
        }
        }
    class Program {
static void Main (string [] args)
    {
    try
      {
Console.WriteLine ("Inside first level try block.");
   LevelOneMethod ( );
      }
   catch (CustomException ex)
        {
Console.WriteLine ($"Caught Custom Exception at first level:
 {ex.Message}");
      }
     catch (Exception ex)
       {
Console.WriteLine ($"Caught Exception at first level : {ex.Message}");
        }
```

```
Console.WriteLine("In End of program.");
        }
static void LevelOne Method ()
        {
    try
     {
Console.WriteLine("Inside Second level try block.");
Level Two Method ( );
        }
   catch (Custom Exception ex)
        {
Console.WriteLine ($ "Caught Custom Exception at second level:
 {ex.Message}");
        }
    catch (Exception ex)
        {
Console.WriteLine ($"Caught Exception at second level : {ex.Message}
                "");
                }
                }
   Static void Level Two Method ( )
        {
    try {
Console.WriteLine("Inside third level try block.");
throw new CustomException ("Custom exception occurred");
        }
   catch (CustomException ex)
     {
Console.WriteLine ($"Caught Custom Experience at third level:
   {ex.Message}");
        }
```

```
catch (Exception ex)
    {
Console.WriteLine ($"Caught Exception at third level: {ex.Message}"
                                                        );
    }
}
}
```

LAB-4

Write a program to insert, update and delete a record of student into a database in a windows form based application.

A program to insert, update and delete a record of student into a database in a windows form based application is:

Source code:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
 namespace StudentDatabaseApp {
  public partial class MainForm: Form
     {
  private SqlConnection connection;
  private string connectionString = "your_connection_string_here";
    public MainForm ( )
      {
    InitializeComponent ( );
connection = new SqlConnection (connectionString);
        }
private void MainForm_Load (object, sender, EventArgs e)
        {
        LoadData ( );
         }
private void LoadData ( )
        {
      try {
 string query = "SELECT * FROM Student";
SqlDataAdapter adapter = new SqlDataAdapter (query,
                                    connection);
```

```
DataTable table = new DataTable();
  adapter.Fill(table);
dataGridViewStudents.DataSource = table;
        }
    catch (Exception ex)
        {
MessageBox.Show("Error: " + ex.Message);
        }
        }
private void btnInsert_Click(object sender, EventArgs e)
        {
    try
        {
string query = "INSERT INTO Students (Name, Age, Grade) Values
   VALUES (@Name, @Age, @Grade)";
SqlCommand command = new SqlCommand(query, connection);
 command.Parameters.AddWithValue("@Name", txtName.Text);
 command.Parameters.AddWithValue("@Age", Convert.ToInt32
    (txtAge.Text));
 command.Parameters.AddWithValue("@Grade", txtGrade.Text);
      connection.Open();
      command.ExecuteNonQuery();
      connection.Close();
    LoadData();
        }
    catch (Exception ex)
        {
MessageBox.Show("Error: " + ex.Message);
        }
        }
```

```
Private void btnUpdate_Click (object sender, EventArgs e)
        {
    try {
String query = "UPDATE Students SET Name = @Name, Age =
@Age, Grade = @Grade, WHERE StudentID = @StudentID";
Sql Command command = new Sql Command (query, connection);
command. Parameters. Add With Value ("@Name", txtName. Text);
command. Parameters. Add With Value ("@Age", Convert. To Int 32
 (txtAge. Text));
command. Parameters. Add With Value ("@Grade", txtGrade. Text);
Command. Parameters. Add With Value ("@Student ID", data Grid View-
Students. Current Row. Cells ["Student ID"]. Value);
    Connection. Open ( );
    command. Execute Non Query ( );
    connection. Close( );
     Load Data ( );
        }
     catch (Exception ex)
        {
Message Box. Show ("Error :" + ex. Message);
        }
        }

private void btnDelete_Click (object sender, Event Args e)
        {
    try {
String query = "DELETE FROM Students WHERE StudentID =
                @StudentID";
Sql Command command = new Sql Command (query, connection);
```

```
command.Parameters.AddWithValue("@StudentID", dataGridViewStudents.CurrentRow.Cells["StudentID"].Value);
    connection.Open();
    command.ExecuteNonQuery();
    connection.Close();
    LoadData();
    }
catch (Exception ex)
    {
MessageBox.Show("Error: " + ex.Message);
        }
    }
}
```

LAB - 5

Write a program to show list of employee in web form application and filter it using employee name, contact no and email address.

A program to show list of employee in web form application and filter it using employee name, contact no and email address is:

Source code:

```
using System;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

namespace EmployeeManagement
    {
public partial class EmployeeList : System.Web.UI.Page
    {
protected void Page_Load (object sender, EventArgs e)
    {
 if (!IsPostBack)
    {
   BlindEmployeeData ( );
    }
    }
protected void BindEmployeeData ( )
    {
string constr = ConfigurationManager.ConnectionStrings
 ["ConnectionString"].ConnectionString;
using (SqlConnection con = new SqlConnection (constr))
    {
```

```
using (SqlCommand cmd = new SqlCommand ("SELECT * FROM Emplo-
yees", con))
    {
  con.Open ( );
SqlDataAdapter da = new SqlDataAdapter (cmd);
  DataTable dt = new DataTable ( );
  da.Fill (dt);
gvEmployees.DataSource = dt;
gvEmployees.DataBind ( );
      }
      }
      }
protected void btnFilter_Click (object sender, EventArgs e)
    {
string filterName = txtFilterName.Text;
string filterContact = txtFilterContact.Text;
string filterEmail = txtFilterEmail.Text;

   string query = "SELECT * FROM Employees WHERE 1 = 1";
if (!string.IsNullorEmpty (filterName))
  query += "AND Name LIKE '%'" + filterName + "%' ";
if (!string.IsNullorEmpty (filterContact))
  query += "AND ContactNumber Like '%." + filterContact +
                    "%.'";
if (!string.IsNullorEmpty (filterEmail))
  query += "AND EmailAddress LIKE '%." + filterEmail +
                    "%.'";
```

```
string constr = ConfigurationManager.ConnectionStrings["
    ConnectionString"].ConnectionString;
using (SqlConnection con = new SqlConnection (constr))
      {
  using (SqlCommand cmd = new SqlCommand (query, con))
      {
    con.Open ( );
SqlDataAdapter da = new SqlDataAdapter (cmd);
   DataTable dt = new DataTable ( );
    da.Fill (dt);
 gvEmployees.DataSource = dt;
 gvEmployees.DataBind ( );
          }
           }
          }
          }
          }
```