LAB SESSION 10

Controls in C#

Objective:
To learn about use of different controls like buttons, text boxes, combo box and timer controls in C# applications and how to apply graphics and Mouse events.

Introduction:

Example:
Create a simple C# that will place one combo box filled with the names of different shapes by using item property of combo box and application should be able to draw the selected shape of combo box control by creating graphic ,pen and solid brush objects.

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

      private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
      {
         Graphics g = base.CreateGraphics();
         Pen p = new Pen(Color.Blue);
         SolidBrush sb = new SolidBrush(Color.Brown);
         g.Clear(Color.White);

         switch (comboBox1.SelectedIndex)
         {
            case 0:
               g.FillEllipse(sb, 50, 50, 150, 150);
               break;
            case 1:
               g.DrawRectangle(p, 50, 50, 150, 150);
               break;
            case 2:
               g.DrawEllipse(p, 50, 80, 150, 170);
               break;
            case 3:
```

```
                g.DrawRectangle(p, 50, 50, 50, 50);
                break;
            }

        }

    }
}
```

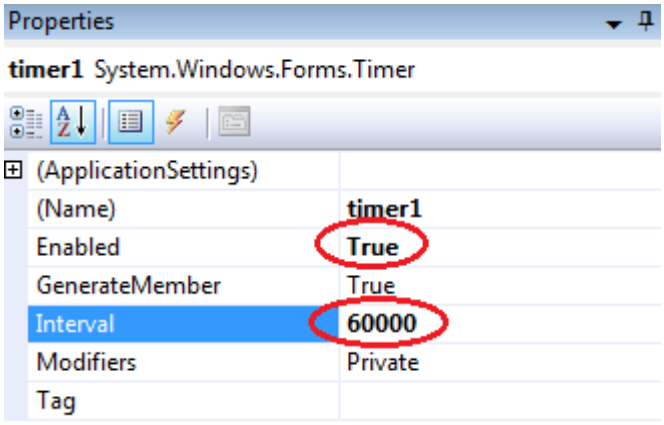
Timer Control:
he *Timer Control* plays an important role in the development of programs both Client side and Server side development. A **Timer** control raises an event at a given interval of time without using a secondary thread. If you need to execute some code after certain interval of time continuously, you can use a timer control.

Timer Class Properties

Enabled property of timer represents if the timer is running. We can set this property to true to start a timer and false to stop a timer.

Interval property represents time on in milliseconds, before the Tick event is raised relative to the last occurrence of the Tick event. One second equals to 1000 milliseconds. So if you want a timer event to be fired every 5 seconds, you need to set Interval property to 5000.

Timer t = new Timer();
Namespace: System.Timers
Assembly: System (in System.dll)



Task 2:
Create an array of Buttons that would display buttons dynamically at run time.
Sample Code:

```
private void Form1_Load(object sender, EventArgs e)
    {
      Button[] b = new Button[5];

      for (int i = 1; i <= b.Length-1; i++)
      {
        b[i] = new Button();
        b[i].Text = "Button" + i;
        b[i].Size = new Size(70,  30);
        b[i].Location = new Point(i + 70, i + 100);
        b[i].Left = 50*i;
        b[i].Top = i * 70;
        b[i].Click += new System.EventHandler(button1_Click);
        this.Controls.Add(b[i]);


      }
    }
```

Task 3:
Create a slideshow of pictures by using picture box control and timer control. Make a folder on any drive containing all of your pictures, you want to include in a slideshow. Rename all the pictures starting with 1.jpg, 2.jpg… n.jpg.

Task 4:
Using Mouse up, down and Mouse move events, Create an application that can draw freehand drawing.

Sample Code:

```
        int readypaint = 1;

    private void Form1_Load(object sender, EventArgs e)
    {
       timer1.Start();
    }

    private void Form1_MouseUp(object sender, MouseEventArgs e)
    {
       readypaint = 0;
    }

    private void Form1_MouseMove(object sender, MouseEventArgs e)
    {
       if (readypaint == 1)
       {
          Graphics g = CreateGraphics();
          Pen p = new Pen(Color.Red);
          g.DrawEllipse(p, e.X, e.Y, 10,10);
       }
    }

    private void Form1_MouseDown(object sender, MouseEventArgs e)
    {
       readypaint = 1;
    }
```

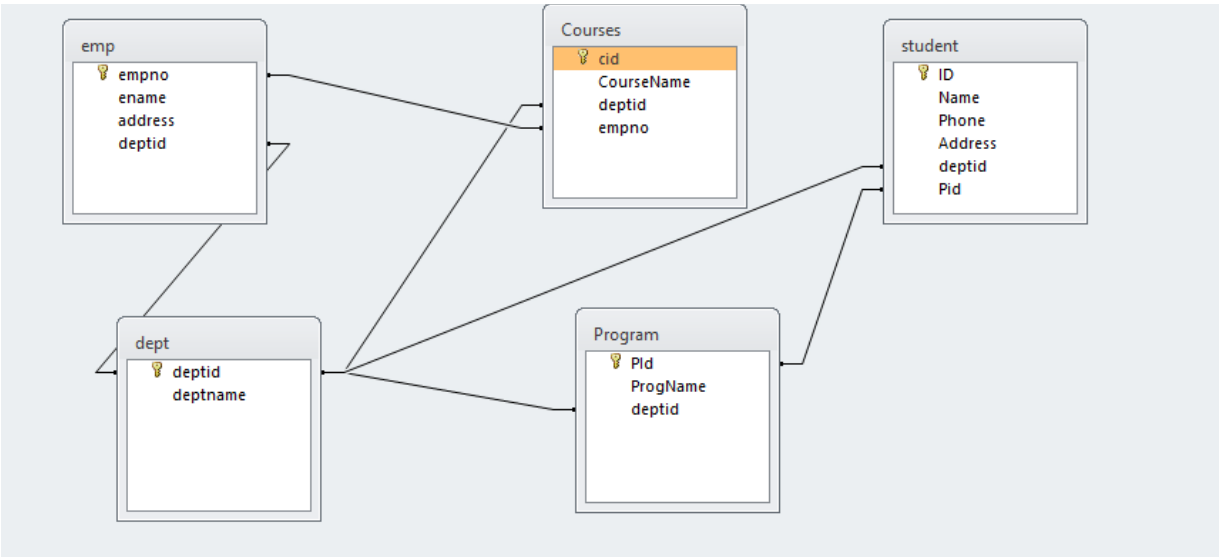
The main objective of this lab is to be able to use databases C# applications using **ADO.Net** classes and **Microsoft SQL Server and Microsoft Access**. We will use data access components tools to connect to, retrieve and update data in database. We will explore how we can leverage the built-in capabilities of ADO .NET to extract and manipulate data as well as insert, update and delete data in SQL Server. With this, we will take a look at **CurrencyManager** object to navigate the records in our bound controls.

Program 1
This simple application how to create database connectivity in C # application.

Steps

Create a Database consisting of four tables containing necessary attributes of these tables along with the connectivity's .

1. Department.
2. Teachers.
3. Student.
4. Subjects
5. Courses
6. Program



1. Start a new Windows Application project. Design GUI as shown below:

2. As this in untyped connection, we have to import System.Data.Oledb name space
3. We'll declare connection, Data Adapter, Data Set and an integer variable as globel:

```
OleDbConnection con = new OleDbConnection("Provider=Microsoft.jet.Oledb.4.0;Data Source= e:\SalesDatabase.mdb;");
 OleDbDataAdapter adap = new OleDbDataAdapter("select * from student", "Provider=Microsoft.jet.Oledb.4.0;Data Source=e:/std.mdb;");
  DataSet d = new DataSet("student");

  Int intCurrentIndex =0;
```

1. Data Adapter acts as a bridge between application and connection. For that a Data Adapter must know which database and what table(s) to use as we'll use Data Adapter to fill our Data Set: On form's load event, write following line of code:
2.

```
da.Fill(ds);
'DS is the name of Data Set
```

Do It Yourself: Code click events of **"<"** button and **">>"** button yourself.

Program 2
This simple application demonstrates How to Use DataGridView control to display data:

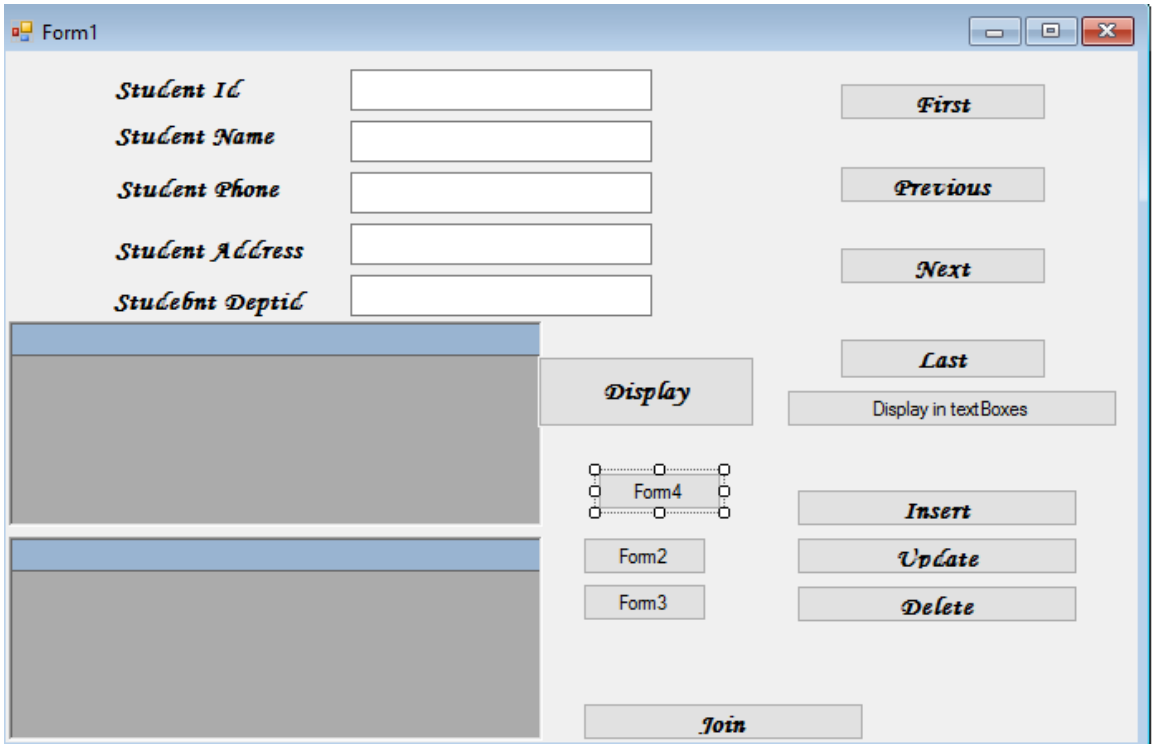
1. Open a new form and place DataGrid Control and a Button on your form.
2. As in previous example import Oledb namespace and declare Connection, Data Adapter and Data Set as global variables.
3. Go to Load Event of your for and enter the following coding:

```
string constr = "Provider=Microsoft.Jet.Oledb.4.0; Data source =e:/std.mdb";
OleDbConnection con = new OleDbConnection(constr);
        string strsql = "select * from student";
OleDbDataAdapter adap = new OleDbDataAdapter(strsql, con);
DataSet d1 = new DataSet("student");
con.Open();
adap.Fill(d1, "student");
dataGrid1.DataSource = d1;
```

4. Save and execute your work.

Program 3:

Enhance your form according to snapshot:(Data Manipulation)



Solution:

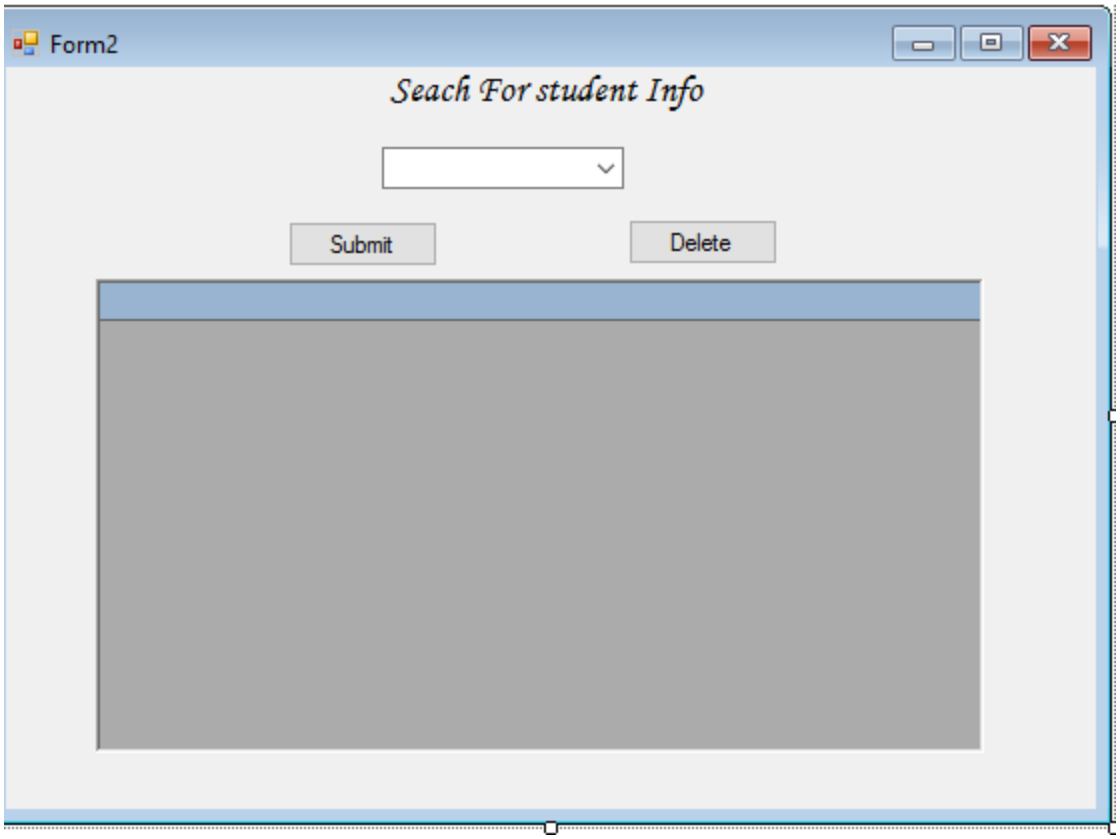
```
OleDbConnection con = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;
Data source=f:/std.mdb");
    DataSet d1 = new DataSet("student");
    int counter = 0;
    private void button1_Click(object sender, EventArgs e)
    {


    }

    private void button6_Click(object sender, EventArgs e)
    {

      textBox1.Text = d1.Tables[0].Rows[0]["ID"].ToString();
      textBox2.Text = d1.Tables[0].Rows[0]["Name"].ToString();
      textBox3.Text = d1.Tables[0].Rows[0]["Phone"].ToString();
      textBox4.Text = d1.Tables[0].Rows[0]["Address"].ToString();
      textBox5.Text = d1.Tables[0].Rows[0]["deptid"].ToString();


    }
```

```
private void Form1_Load(object sender, EventArgs e)
{
   con.Open();

}

private void button2_Click(object sender, EventArgs e)
{
   if (counter > 0)
   {
      counter = 0;

      textBox1.Text = d1.Tables[0].Rows[counter]["ID"].ToString();
      textBox2.Text = d1.Tables[0].Rows[counter]["Name"].ToString();
      textBox3.Text = d1.Tables[0].Rows[counter]["Phone"].ToString();
      textBox4.Text = d1.Tables[0].Rows[counter]["Address"].ToString();
      textBox5.Text = d1.Tables[0].Rows[counter]["deptid"].ToString();

   }
}

private void button3_Click(object sender, EventArgs e)
{
   if (counter > 0)
   {
      counter = counter - 1;
      textBox1.Text = d1.Tables[0].Rows[counter]["ID"].ToString();
      textBox2.Text = d1.Tables[0].Rows[counter]["Name"].ToString();
      textBox3.Text = d1.Tables[0].Rows[counter]["Phone"].ToString();
      textBox4.Text = d1.Tables[0].Rows[counter]["Address"].ToString();
      textBox5.Text = d1.Tables[0].Rows[counter]["deptid"].ToString();
   }
   else
   {
      MessageBox.Show(" U are lready on the first record");
   }
}

private void button4_Click(object sender, EventArgs e)
{
   if (counter < d1.Tables[0].Rows.Count - 1)
   {
      counter = counter + 1;
      textBox1.Text = d1.Tables[0].Rows[counter]["ID"].ToString();
      textBox2.Text = d1.Tables[0].Rows[counter]["Name"].ToString();
```

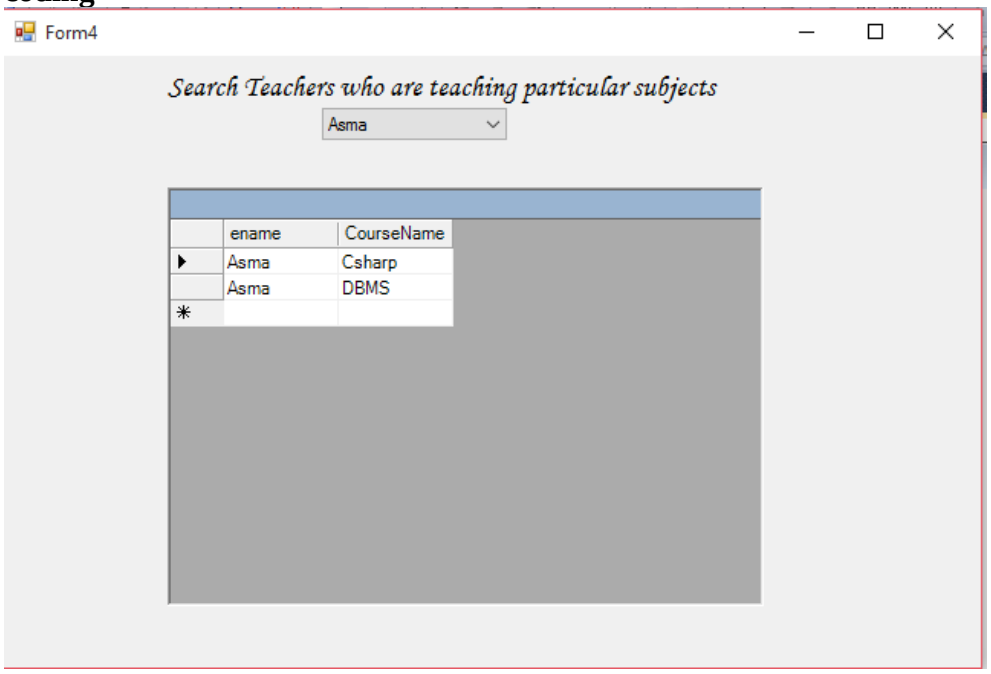
```
            textBox3.Text = d1.Tables[0].Rows[counter]["Phone"].ToString();
            textBox4.Text = d1.Tables[0].Rows[counter]["Address"].ToString();
            textBox5.Text = d1.Tables[0].Rows[counter]["deptid"].ToString();
        }
    }

    private void button5_Click(object sender, EventArgs e)
    {
        if (counter < d1.Tables[0].Rows.Count - 1)
        {
            counter = d1.Tables[0].Rows.Count - 1;
            textBox1.Text = d1.Tables[0].Rows[counter]["ID"].ToString();
            textBox2.Text = d1.Tables[0].Rows[counter]["Name"].ToString();
            textBox3.Text = d1.Tables[0].Rows[counter]["Phone"].ToString();
            textBox4.Text = d1.Tables[0].Rows[counter]["Address"].ToString();
            textBox5.Text = d1.Tables[0].Rows[counter]["deptid"].ToString();
        }
    }

    private void button7_Click(object sender, EventArgs e)
    {

        OleDbDataAdapter adap1 = new OleDbDataAdapter(" Select student.ID , dept.deptid from student inner join dept on dept.deptid=student.deptid ", con);
        DataSet d2 = new DataSet();
        adap1.Fill(d2,"student");
        dataGrid1.DataSource = d2;
        OleDbDataAdapter adap2 = new OleDbDataAdapter("select Courses.CourseName,emp.ename from emp inner join courses on emp.empno=courses.empno", con);

       adap2.Fill(d2,"emp");
        dataGrid2.DataSource=d2;


    }

    private void button8_Click(object sender, EventArgs e)
    {
        OleDbCommand com1 = new OleDbCommand("Insert into student(ID,Name,Phone,Address,deptid) values(' " + textBox1.Text + " ',' " + textBox2.Text + " ',' " + textBox3.Text + " ',' " + textBox4.Text + " ',' " + textBox5.Text + " ')", con);
        com1.ExecuteNonQuery();
        MessageBox.Show("One record has been added");
```

```
        }

        private void button10_Click(object sender, EventArgs e)
        {
            OleDbCommand com2 = new OleDbCommand("DELETE FROM student where
ID = '" + textBox1.Text + " '", con);
            com2.ExecuteNonQuery();
            MessageBox.Show("One record has been deleted ");


        }

        private void button11_Click(object sender, EventArgs e)
        {

            Form2 f2 = new Form2();
            f2.Show();
        }

        private void button9_Click(object sender, EventArgs e)
        {
            OleDbCommand com3 = new OleDbCommand("UPDATE student set
Address='Lahore' where ID= '" + textBox1.Text + " ' ", con);

            com3.ExecuteNonQuery();
            MessageBox.Show(" One record has been updated");


        }

        private void button12_Click(object sender, EventArgs e)
        {
            Form3 f3 = new Form3();
            f3.Show();

        }

        private void button13_Click(object sender, EventArgs e)
        {
            Form4  f4 = new Form4();
            f4.Show();
        }
    }
}
```

Task 1: Design this interface for second Form.



Task 2: **Design this interface for third form And write down necessary coding**



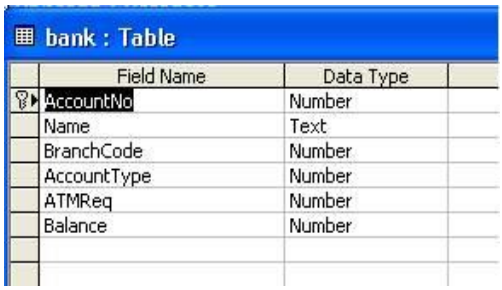
Program 5:

This database application demonstrates the use of ADO.Net classes using Access Database.Steps:

1. For this example we have created MS-ACCESS database named **Csharp1.mdb**. The table structure for the **bank** table used in this example is show below:

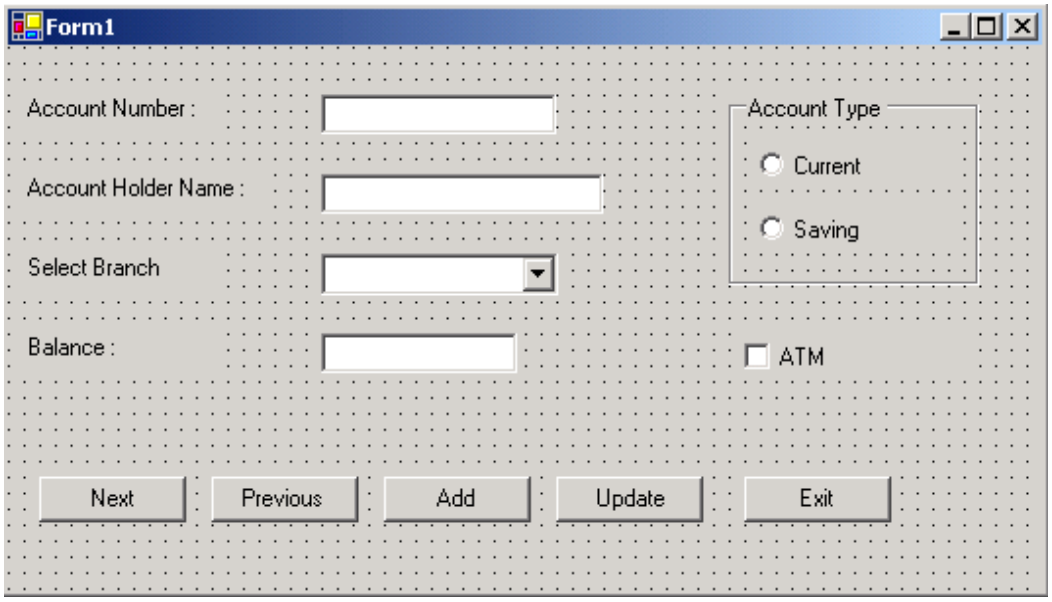
   The properties of the fields are as follows

   - AccountNo - Number - Long Integer
   - Name - string
   - BranchCode - int
   - AccountType - int
   - ATMReq - int
   - Balance - Number - Decimal



2. Create a new Windows Application project. From toolbox's Data tab, select OleDbDataAdapter component and configure it to establish connection with your database.

3. Design your GUI as shown below:



4. In this application, we will be working with individual data bound controls.

5. Select first Textbox control on your form.

6. Open **DataBinding** section on control's Properties window, locate the Text item, and expand its list of possible settings. Select the one you want to display in your control as shown below: