

Name: **Session:**

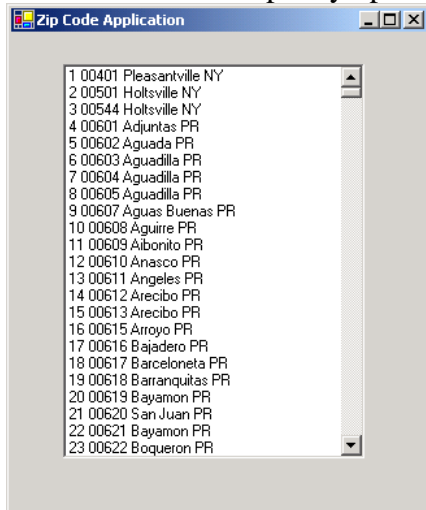
Programming II

Lab Exercise 5/11/2022

Zip Code Application

In this lab you will build a ZIP code lookup application. On the server in the //Ada/Data Files/ZipCode Database folder you will find an Access database called US-Cities.mdb which contains the Zip codes for all 42,149 cities in the United States as well as a text file called states.txt. **Do not forget to put these files in your project folder for them to be accessed properly.**

We will start off simple by opening the database and loading it into a Listbox.



This will ensure that you are getting the database into your program. To make things simpler, I created a City class to hold the fields of the database. I then created a List class object to store the cities in.

Here is the code for the City class:

```
class City
{
    private string rid;
    private string zipCode;
    private string city;
    private string state;

    public City(string id, string zc, string cty, string st)
    {
        rid = id;
        zipCode = zc;
        city = cty;
        state = st;
    }
}
```

1. Create the following variables in the Form_Load procedure

```
//Create variables
```

```
string sql, id, zip, city, state;
```

```
//Create a connection string variable
```

```
string sConnection;
```

```
//Setup database connection
```

```
OleDbConnection dbConn = new OleDbConnection();
```

```
OleDbCommand dbCmd = new OleDbCommand();
```

```
OleDbDataReader dbReader;
```

```
//Create City object
```

```
City s;
```

2. Call the List constructor construct an List object to hold cities

```
List<City> cities = new List<City>();
```

3. Define the database connection string

```
sConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source = US-Cities.mdb";
```

4. Define the SQL string

```
sql = "Select * From CityStateZip Order by State Asc, City Asc;";
```

5. Create and open the database connection

```
dbConn = new OleDbConnection(sConnection);
```

```
dbConn.Open();
```

6. Create and open the database command object

```
dbCmd = new OleDbCommand();
```

```
dbCmd.CommandText = sql;
```

```
dbCmd.Connection = dbConn;
```

7. Start the database reader

```
dbReader = dbCmd.ExecuteReader();
```

8. Read in the database records, create a city object, add the city object to the List, and add database record to the Listbox.

```
while (dbReader.Read())
{
    //Read the database fields and convert to strings
    id = System.Convert.ToString(dbReader.GetValue(0));
    zip = System.Convert.ToString(dbReader.GetValue(1));
    city = System.Convert.ToString(dbReader.GetValue(2));
    state = System.Convert.ToString(dbReader.GetValue(3));

    //Construct a City object
    s = new City(id, zip, city, state);

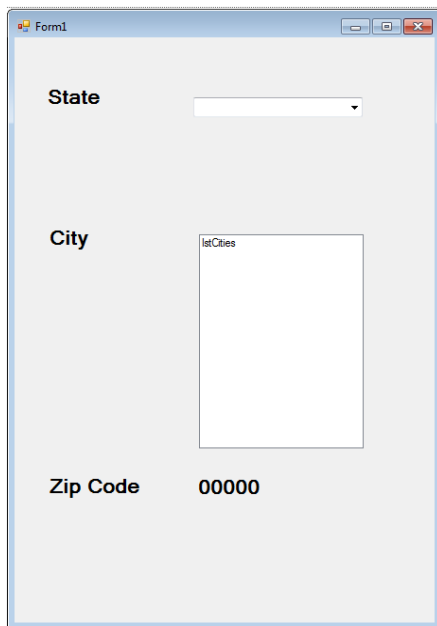
    //Add City object to List
    cities.Add(s);

    //Add field variables to ListBox
    lstCities.Items.Add(id + "    " + zip + "    " + city + "    " + state);
}
```

9. Close the database reader as well as the database connection

```
dbReader.Close();
dbConn.Close();
```

Once you have successfully loaded the database, we will make an improved application to allow us to have a nicer user interface.



The screenshot shows a Windows application window titled "Form1". Inside the window, there are three main components: a "State" label followed by a dropdown menu, a "City" label followed by a list box (the list box has "lstCities" written in its top-left corner), and a "Zip Code" label followed by a text box containing the value "00000".

In the interface you would select the state and then go through the List and if the state matches, add that city to the Listbox. When you select the item from the Listbox, you would then go through the List to find the city/state match and retrieve the zip code when the match occurs. You will need to modify your code to get this to work.

1. Add three functions to your City class

```
public string getState()
{
    return state;
}

public string getCity()
{
    return city;
}

public string getZIP()
{
    return zipCode;
}
```

2. Create two Lists that are global to Form1

```
List<City> cities = new List<City>();
List<string> zips = new List<string>();
```

3. Modify the Form1_Load function as such:

```
//Create variables
string sql, id, zip, city, state;

//Create a connection string variable
string sConnection;

//Setup database connection
OleDbConnection dbConn = new OleDbConnection();
OleDbCommand dbCmd = new OleDbCommand();
OleDbDataReader dbReader;

//Create City object variable
City s;

//Create the connection string
sConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source = US-Cities.mdb";
```

```

//Define the SQL query ordering by State then City ascending order
sql = "Select * From CityStateZip Order by State Asc, City Asc;";

//Connect to the database
dbConn = new OleDbConnection(sConnection);
dbConn.Open();

//Create the database command object
dbCmd = new OleDbCommand();

//Assign the SQL command to the database command object
dbCmd.CommandText = sql;

//Connect the SQL command to the database
dbCmd.Connection = dbConn;

//Start reading the database
dbReader = dbCmd.ExecuteReader();

//Start reading database records
while (dbReader.Read())
{
    //Read the database fields and convert to strings
    id = System.Convert.ToString(dbReader.GetValue(0));
    zip = System.Convert.ToString(dbReader.GetValue(1));
    city = System.Convert.ToString(dbReader.GetValue(2));
    state = System.Convert.ToString(dbReader.GetValue(3));

    //Construct a City object
    s = new City(id, zip, city, state);

    //Add City object to ArrayList
    cities.Add(s);

    //Add field variables to ListBox
    lstCities.Items.Add(id + "    " + zip + "    " + city + "    " + state);
}

//Close connection to database
dbReader.Close();
dbConn.Close();

```

```
//Read in the states to the comboBox collection
StreamReader sr = new StreamReader("states.txt");
while (sr.Peek() >= 0)
{
    cboStates.Items.Add(sr.ReadLine());
}
sr.Close();
```

Note: In order to use StreamReader you must add using System.IO;

4. Create a function for the ComboBox cboStates_SelectedIndexChanged event

```
//Create variable to hold name of state
string theState;

//Clear the cities ListBox
lstCities.Items.Clear();

//Get the state from the ComboBox
theState = cboStates.SelectedItem.ToString();

//Extract the state abbreviation
theState = theState.Substring(0, 2);

//Check each city in the cities List
for (int i = 0; i < cities.Count; i++)
{
    //if the city is in the state then add the city to lstCities
    //and add its zipcode to the zips List
    if (cities[i].getState() == theState)
    {
        lstCities.Items.Add(cities[i].getCity());
        zips.Add(cities[i].getZIP());
    }
}
```

5. Create a function for the Listbox lstCities_SelectedIndexChanged event

```
//Add the selected city zipcode to the ZIP code label
lblZip.Text = zips[lstCities.SelectedIndex].ToString();
```

6. Now test your program. When your program is working, create a screenshot of your working program showing the zipcode of some town that has an interesting name and submit it.