**Lab week 4 :**

**Aim : Creating a simple class library and using it with simple Widows Form APP.**

Task A:

1. Create a new class library project in VS 2017.

Select file->new-Project-class library

Class library is used to write “reusable code”. Which can be reused  in a console app, in a desktop app, in a web app, or all three.

1. Name the class library “PolytechLibrary” and name the solution “PolytechApp”.

It makes sense to have different names for solution and other projects as a solution can have multiple projects. Like we will be adding one in this lab and a few more in other labs.

1. Create a class name it “Polytech”. Right click the project

Select add->class

You may rename the default class ”class1” which was created when you created the library to “Polytech” or just delete the “class1”

1. Add following f*ield*s to the newly created Polytech class

*Name*

*Address*

*City*

*Region*

*Postcode*

*PhoneNumber*

*twitterAddress*

select appropriate *datatype* and *access specifiers* for each of the above.

1. Add auto properties for fields

*Name, Address, City, Region, PhoneNumber*

For *twitterAddress* field , add a property that checks if the *twitterAddress* *value* starts with symbol ‘@’

If not then throws an exception

*throw new Exception(“The titter address must begin with @”);*

1. Add a default and a parametrized constructor.
2. *Override* a *tostring()* method*.*

public override string ToString( ) {

var sb = new StringBuilder();

sb,Appendline(Name);

sb.Appendline(Address);

//similary add all the fields;

return sb.Tostring()

}

Class library cannot be execute alone. We need another application/project to test the library.

Now let’s use this library in some application. As, stated earlier class library can be used with many other application. In this case lets create a windows form application. Windows Form Application are a bit older concept but are easy to use . Windows form application is same as WPF except that WPFs can be deployed into windows store. WPF require knowledge of XAML which is another language to learn - but we will look at a simple WPF app in next lab along with inheritance and polymorphism.

1. Create a Windows Form App and add Polytech class to it. Right click solution

* Select->add->new Project->
* Select->Windows Classic Desktop, on the left pane then
* Select->Windows form App

Name it “PolytechFormApp”

1. Add a reference or create the dependency to the class library we created earlier.

Right click on the *References* underneath “ PolytechFormApp”

* Select add Reference
* In the left pane click on projects -> solution
* Check “PolytechLibrary” on the right pane -> click OK.

Now that you have added a reference to the class library select start-up project. You can see your class library “PolytechLibrary” is highlighted in bold in the solution explorer. You will need to change the class library itself doesn’t execute.

1. Right click on “PolytechLibrary” select -> set as start-up project.

**Create a simple GUI**

1. Inside “PolytechFormApp” find “Toolbox” on the right hand side.

Select “All Windows Forms”, you can see all the controls.

* Drag a few ( 7 of them) “labels” on to the GUI form.
* Drag a few (7 of them ) textboxes and place them next to each of the labels.
* Click the first label and find the “Text” attribute for it in the “properties” window usually opened in the right bottom of the screen. If not then open it from “view” in the toolbar.

Change the value of “Text” attribute from “label” to “Name”

{the first field name in Polytech class}

Similarly do it for all other labels- changes the Text attribute the fields name from Polytech class(address, city …).

* Add a button from Toolbox to the field , name it “ok” button by going to its property window.
* Similarly, change the “Name” attribute of each of the “textboxes” to a more meaningful name.

For ex: textbox1 to txtName

textbox2 to txtAdress

and so on..

**Working with user inputs.**

1. Double click the button of the form, You should able to see code(a method) behind it( in form1.cs) something like :

private void ok\_click (object sender , EvenArgs e)

{

}

1. Create an instance of the “Polytech” class inside the button handler method and add the following

private void ok\_click (object sender , EvenArgs e)

{

Var testpolytech= new Polytech( );

//now you can assign all field values from user inputs, like this

testpolytech.Name= txtName.Text;

testpolytech.Adress= txtAddress.Text;

//similarly add for each fields….

//for twitteradress filed add something like this. Ask Manish why??

Try

{

Testpolytech.TwitterAddress= textTwitter.Text;

}catch (exception ex){

MessageBox.Show(ex.message);

}

}

1. Using the Tostring() override from the Polytech class. Add a line as follows at the end of form1.cs.

private void ok\_click (object sender , EvenArgs e)

{

Var testpolytech= new Polytech( );

//now you can assign all field values from user inputs, like this

testpolytech.Name= txtName.Text;

testpolytech.Adress= txtAddress.Text;

//similarly add for each fields….

//for twitteradress filed add something like this. Ask Manish why??

Try

{

Testpolytech.TwitterAddress= textTwitter.Text;

}catch (exception ex){

MessageBox.Show(ex.message);

}

MessageBox.Show(testpolytech.ToString());

}

1. Run the program. Input values in the text boxes. Click ok button.

**Try out different controls from the toolbox**

**We will continue this lab in another Lab session. Please save it somewhere where you can get it back.**

Credits: C# essentials, Bruce Van Horn.