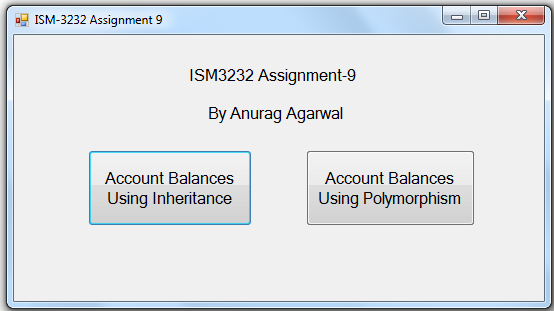
**ISM-3232**

**Assignment-9**

**Inheritance and Polymorphism**

**Due 11/13/14**

This assignment has two parts. In part A you will apply the knowledge of Inheritance and in Part B you will use Polymorphism to perform similar tasks. You will create one form with an appropriate name. This form has two buttons, one for part A, i.e. doing the task using Inheritance and the other for Part B. The project should be called ISM3232Assign9LastNameFirstName.



Part A:

Create a parent class called clsAccount with four properties, namely AccountName (String), DebitBalance (Single), CreditBalance (Single) and NetBalance (Single) and a method named ComputeBalance. This class should be an abstract class, i.e., it should be declared as a MustInherit class. The method ComputeBalance in the parent class should be an Overridable method which computes the NetBalance as DebitBalance minus CreditBalance.

The three sub classes will be clsAssetAccount, clsLiabilityAccount and clsEquityAccount. They will each inherit the properties and methods of the parent class.

The clsLiabilityAccount and clsEquityAccount classes should override the ComputeBalance method as NetBalance = CreditBalance minus DebitBalance

clsAssetAccount will not override the ComputeBalance method.

When the first button on the form is clicked the following should happen:

Open an output file called AccountBalancesUsingInheritance.txt

Instantiate four objects – Cash, AR, AP and RE as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | Class | AccountName | DebitBalance | CreditBalance |
| Cash | clsAssetAccount | Cash | 10000 | 4000 |
| AR | clsAssetAccount | AccountsReceivable | 5000 | 2000 |
| AP | clsLiabilityAccount | AccountsPayable | 6000 | 8000 |
| RE | clsEquityAccount | RetainedEarnings | 7000 | 14000 |

After instantiating each of these objects, run the ComputeBalance method for each object and write to the output file, the AccountName and the NetBalance. The output file should look like this:

Cash,6000  
AccountsReceivable,3000  
AccountsPayable,2000  
RetainedEarnings,7000

Close the output file and display the message “Done”

Part B:

Create an interface called IntAccount having four properties and a method. The properties are called AccountName, DebitBalance, CreditBalance and NetBalance. The method is called ComputeBalance.

Create three classes that will implement the interface created above. The three classes are IAssetAccount, ILiabilityAccount and IEquityAccount.

These three classes will implement the four properties and the method of the interface.

In the class IAssetAccount, the ComputeBalance will compute NetBalance as DebitBalance minus CreditBalance. In the other two classes, it will be the reverse.

When the button for part B is clicked the following should happen:

Open an output file called AccountBalancesUsingPolymorphism.txt

Declare an object named Account as the interface name.

It should then instantiate this object (Account) as the IAssetAccount class and assign the AccountName as “Cash” and DebitBalance and CreditBalance for the cash account as in the table. It should then compute the NetBalance. It should then write the AccountName and the NetBalance in the output file.

It should then instantiate this object (Account) as the IAssetAccount class and assign the AccountName as “AccountsReceivable” and DebitBalance and CreditBalance for the AR account as in the table. It should then compute the NetBalance. It should then write the AccountName and the NetBalance in the output file.

It should then instantiate this object (Account) as the ILiabilityAccount class and assign the AccountName as “AccountsPayable” and DebitBalance and CreditBalance for the AP account as in the table. It should then compute the NetBalance. It should then write the AccountName and the NetBalance in the output file.

It should do the same for the RE account.

Close the output file and display the message “Done”.

The output file should be identical to the one created in part A.