Homework OOP

1. Create a Currency class that has:

- Constructors

With two int parameters - which show dollars and cents

With one parameter double

No parameters - initialize with zeros.

- Properties

Int Dollars - total amount of dollars (read-only)

Int Cents - The whole number of cents (read-only)

- Methods

double ToDoublel () - returns the double value

Currency Add (Currency currency) - return a new instance of Currency as the sum of the current and the currency parameter.

Currency Multiply (int multiplier) - return a new instance of Currency that has the value of the current multiplied by the number

string ToString () - rotate in the format "D.CC" (for example - 0.20, 20.75, etc.)

(override the ToString () method from the Object)

2. Person's Class - has a name and age. The Citizen subsidiary(inherited from Person), that implements the IHasId interface - has an identifier code - the string Id {get;} property.

Another class - LegalPerson (legal entity) - also implements the IHasId interface, but is not inherited from Person.

Class Bank - has methods

void PutMoneyToAccount (IHasId accountHolder, int quantity) - put money on the account

void GetMoneyFromAccount (IHasId accountholder, int quantity) - withdraw money from an account

int GetAccountBalance (IHasId accountHolder) - get an account balance

Check the bank for both Citizen instances and LegalPerson.

3. Create the interface ILogger with the method for outputting the text Log (string). Three implementation of the interface:

ConsoleLogger - Console output

FileLogger - output to file

AggregateLogger - has the ILogger collection, and when calling the Log method calls this method in each of the items in the collection.

(Examples of output to a file can be found here: <https://msdn.microsoft.com/en-us/library/6ka1wd3w(v=vs.110).aspx>)

Make the void LogHappyMessage (ILogger logger) method, which calls the Log method with the text "I am happy!", To check its work on all three implementations.

Homework .NET conventions

Contact your development and automation teams to define:

* Existing conventions (if any);
* How they used in a project;
* Difference between automation teams and development teams conventions (if any).