CEGEP VANIER COLLEGE
Department of Computer Science

Assignment-02: Object Oriented Programming - Inheritance

Assignment-02 weight is 5% of final grade
Max Marks (55)

Course Title : Application Development-1 (Sec-02) **Teacher** : Syed Naseem Afzal
Course Number : 420-331-VA **Due Date** : September 17, 2021

What to hand in?

After successful demonstration, create a compressed file [**Filename**: LastNameFirstNameAssignment02.ZIP], with the following:

a) MS-Word document [**Filename**: LastNameFirstNameAssignment02.docx] containing screen shots of your examples' output window with input values and produced output/result.
b) Your working application project folder(s).

Upload your **LastNameFirstNameAssignment02.ZIP** file in LEA's assignment section.

Note: Your uploaded (ZIP) file should contain all files of the working examples' project folders. Incomplete and / or non-working app's files will not be graded and have zero value.

Person Inheritance Hierarchy — Max Marks (55)

Create an inheritance hierarchy of classes, that a business might use to represent their customers.

1) Design a class **Person** with the following four instance variables: **Max Marks (15)**
   i) one **private `string`** variable to represent the person's last name (e.g. lastName)
   ii) one **private `string`** variable to represent the person's first name (e.g. firstName)
   iii) one **private `string`** variable to represent the person's address (e.g. address)
   iv) one **private `string`** variable to represent the person's phone number (e.g. phone)

2) The **Person** class should provide a constructor with four parameters:
   i) first parameter to receive person's last name and uses it to initialize the private instance variable using **public property `LastName`**.
   ii) second parameter to receive person's first name and uses it to initialize the private instance variable using **public property `FirstName`**.
   iii) third parameter to receive person's address and uses it to initialize the private instance variable using **public property `Address`**.
   iv) fourth parameter to receive person's phone number and uses it to initialize the private instance variable using **public property `Phone`**.

3) The **Person** class should have a public method **`DisplayData`**, that display the person's record by showing the values of four instance variables of the **Person** class object in tabular form as shown below, using C# 6 mechanism "**String Interpolation**" to insert values in string literals to produce formatted output.

PERSON	
Last Name	Laforge-Bourret
First Name	Tsifoina Tahiry
Address	123 Saint-Croix, Montreal, QC
Phone Number	514-123-4567

4) Design a class **Customer**, (derived from the **Person** class) **Max Marks (15)**
Customer class should inherit the functionality of class **Person**. The **Customer** class should have following two additional instance variables:

   i) one **private `int`** variable (e.g. customerNumber) to represent the customer number. The customerNumber instance variable will be used to hold a unique ID for each customer.

   ii) one **private `bool`** variable (e.g. mailingList) to represent the customer's intention. The mailingList variable will be set to **`true`**, if customer wishes to be on a mailing list, or **`false`** if the customer does not wish to be on a mailing list.

5) The **Customer** class should provide a constructor with six parameters:

   i) first four parameters' values will be passed to constructor of base class (**Person**) to initialize the inherited instance variables from the base class (**Person**) by the derived class **Customer**.

   ii) fifth parameter to receive customer number's value and uses it to initialize the private instance variable using **public property `CustomerNumber`**. The **`CustomerNumber`** property should validate the customer number value received via constructor's parameter, to ensure that it is greater than zero (0); if not, ignore the received customer number value, assign 99999 as customer number to the instance variable (customerNumber) and display the message "**Customer number should be a positive integer value**."

   iii) sixth parameter to receive person's intention and uses it to initialize the private instance variable with a bool value (**`true`** OR **`false`**) using **public property `MailingList`**.

6) The **Customer** class should override the inherited method **`DisplayData`** from the base class **`(Person)`**, so that it should display values of six instance variables in tabular form as shown below, using C# 6 mechanism "**String Interpolation**" to insert values in string literals to create formatted output.

CUSTOMER	
Customer Number	9876543
Last Name	Laforge-Bourret
First Name	Tsifoina Tahiry
Address	123 Saint-Croix, Montreal, QC
Phone Number	514-123-4567
Mailing List	Yes

A retail store has a preferred customer plan where customers may earn discounts on all their purchases. The amount of a customer's discount is determined by the amount spent on the customer's purchases in the store.

- When a preferred customer spends $500, he or she gets a 5% discount on all purchases.
- When a preferred customer spends $1,000, he or she gets a 6% discount on all purchases.
- When a preferred customer spends $1,500, he or she gets a 7% discount on all purchases.
- When a preferred customer spends $2,000 or more, he or she gets a 10% discount on all purchases.

7) Design a class **PreferredCustomer**, (derived from the **Customer** class). **Max Marks (20)**
The **PreferredCustomer** class should inherit the functionality of class **Customer**. The **PreferredCustomer** class should have following two additional instance variables:
   i) one **private `decimal`** variable (e.g. purchasesAmount) to hold the total of a customer's purchases.
   ii) one **private `int`** variable (e.g. discountLevel) to represent the customer's discount level. The discountLevel instance variable should be set to the correct discount percentage, according to the store's preferred customer plan.

8) The **PreferredCustomer** class should provide a constructor with seven parameters:
   i) first six parameters' values will be passed to constructor of base class (**Customer**) to initialize the inherited instance variables from the base class (**Customer**) by this derived class.
   ii) seventh parameter to receive customer's purchase amount (**`decimal`**) value and uses it to initialize the private instance variable using **public property `PurchasesAmount`**. The **`PurchasesAmount` property** should validate the customer's purchase amount value received via constructor's parameter, to ensure that it is greater than zero (0); if not, ignore the received purchase amount value, assign zero (0) to the instance variable (purchasesAmount) and display the message "**Customer purchases amount should be a positive value**."
   iii) **PreferredCustomer** class constructor should use the **`DiscountLevel`** property to set the discountLevel instance variable value.

9) The **PreferredCustomer** class should have a public property **`DiscountLevel`**. The property should return an integer (int) value to set the instance variable (discountLevel) value to correct discount percentage, according to the store's preferred customer plan.

10) The **PreferredCustomer** class should override the inherited method **`DisplayData`** from the base class **(Customer)**, so that overridden **`DisplayData`** method should display values of eight instance variables in tabular form as shown below, using C# 6 mechanism "**String Interpolation**" to insert values in string literals to create formatted output.

PREFERRED CUSTOMER	
Customer Number	9876543
Last Name	Laforge-Bourret
First Name	Tsifoina Tahiry
Address	123 Saint-Croix, Montreal, QC
Phone Number	514-123-4567
Mailing List	Yes
Purchase Amount	$2,345.67
Discount Level	10%

11) After designing the above mentioned three classes, write a driver application class. **Max Marks (05)**

12) The driver application class should accept following seven input values via keyboard from the application user:
    i) Customer Number
    ii) Customer's Last Name
    iii) Customer's First Name
    iv) Customer's Address
    v) Customer's Phone Number
    vi) Mailing List ("Y" or "N")
    vii) Customer's Purchases Amount

13) The driver application should creates (instantiates) objects of all three class (**`Person`**, **`Customer`**, and **`PreferredCustomer`**) and tests their **`DisplayData`** methods, by calling the **`DisplayData`** method with objects of each class.