**BCS 450 C# Lab – Events**

***Overview***

Create a project in C# using Visual Studio. You will write classes that use events. Specifically, you will create Company and Customer classes that use events.

***Part 1 – Create the project***

Create a C# console application in Visual Studio. Name the project Lab-Events.

***Part 2 – Create a Company class***

Company class member variables:

* Name:string (auto-implemented property, see below)

Company class events:

* ShipmentArrived. New goods shipment arrived.
* Sale. A new sale has just begun.

Company methods:

* void FireShipmentArrived(string shipmentDescription). This method should fire shipment arrived event. It will need to create an insance of ShipmentArrivedEventArgs and populate it with the passed in shipment description.
* void FireSale(string shipmentDescription). This method should fire sale event. It will need to create an insance of SaleEventArgs and populate it with the passed in sale description.

***Part 3 – Create ShipmentArrivedEventArgs class in Company***

Create a class named ShipmentArrivedEventArgs INSIDE the Company.

ShipmentArrivedEventArgs class member variables:

* ShipmentArrivedDescription:string (auto-implemented property, see below)

ShipmentArrivedEventArgs methods:

* ShipmentArrivedEventArgs (string sd) – One parameter constructor that just sets the shipment description.

***Part 4 – Create SaleEventArgs class in Company***

Create a class named SaleEventArgs INSIDE the Company.

SaleEventArgs class member variables:

* SaleDescription:string (auto-implemented property, see below)

SaleEventArgs methods:

* SaveEventArgs(string sd) – One parameter constructor that just sets the SaleDescription.

***Part 5 – Create a Customer class***

Secretary class member variables:

* Name:string (auto-implemented property, see below)

Customer methods

* void ProcessSaleEvent(Object sender, SaleEventArgs evtArgs). Should just print out the following:
  + The string “Processed sale event”.
  + The name of the customer that handled the event.
  + The sale description (from the evtArgs parameter).
* void ProcessShipmentArrivedEvent(Object sender, ShipmentArrviedEventArgs evtArgs). Should just print out the following:
  + The string “Processed shipment arrived event”.
  + The name of the customer that handled the event.
  + The shipment arrived description (from the evtArgs parameter).

***Part 6 – Main***

Should implement the following specifications:

* Create an instance of Company.
* Create two instances of Customer. One instance should have the name “Jane” and the other should have the name “Mark”.
* Add a loop that shows the following menu and gets a user choice:

Event Menu

----------

1 – Jane subscribe to company shipment arrived event.

2 – Mark subscribe to company shipment arrived event.

3 – Jane subscribe to company sale event.

4 – Mark subscribe to company sale event.

5 – Jane unsubscribe to company shipment arrived event.

6 – Mark unsubscribe to company shipment arrived event.

7 – Jane unsubscribe to company sale event.

8 – Mark unsubscribe to company sale event.

9 – Fire company shipment arrived event.

10 – Fire company sale event.

11 – Exit

Menu choice:

* Add the appropriate code to process each menu item. For example, if the user chooses 1 then the Jane Customer instance should subscribe to the Company shipment arrived event of the Company instance. When firing the events you can hard code in a string to put in the event args instance.

Run the program and test each option. Make sure to fire off events when there are different amounts of subscribers and make sure that the proper customer processes the events.