Objective:
Construct an ActiveX DLL component.

What to do:
1. From the course folder or website, download and unzip the Inventory Test project. This application contains two forms that make use of a simple inventory management component. You will have to write this component before the application will run. Examine the code for these forms carefully. Identify the objects from your component: where and how they are used.
2. Add a new project to the project group of this application and set its type to be ActiveX DLL. Inside this component you will implement class modules as indicated below.
3. CItem. This class module represents an item that can be sold. It should have the following properties:
   Name (String)
   ID (Long)
   Description (String)
   Price (Currency)
   All of these properties should be hidden from the user of the component but available to objects inside the component.
4. CStock. This class module represents the stock on hand for any given inventory item. It has the following public API:
   Properties:
   Count (read): How many of the item we have on hand.
   ItemName (read-only): Name of the associated inventory item
   ItemID (read-only): ID of the associated inventory item
   ItemDescription (read-only): Description of the associated inventory item
   ItemPrice (read-only): Price for the associated inventory item
   In addition, the following API is available inside the component:
   Properties:
   Count (write)
   Item (read-write): The CItem object for this stock entry
   Methods:
   Receive (Count As Integer): Adds to stock count
   Ship (Count As Integer): Removes from stock count
5. CInventory. This is a collection class (generate its outline using the class wizard if you wish) that holds some number of CStock entries. Use the item name as the key.
   Properties:
   Count
   Item(Name As String)
   NewEnum
   Methods:
   AddStock(Name As String, Description As String, Price As Currency, Count As Integer)
   Receive(Name As String, Count As Integer)
   Remove(Name As String)
   Ship(Name As String, Count As Integer)
   Item IDs should be generated automatically by AddStock. Use the example on page 365 of the book. In addition, the CInventory class should raise an “OutOfStock” event whenever the stock of some item falls to zero. This event is handled by the Inventory Test application – look for gInventory_OutOfStock.
What to turn in:
Before class time, place your project group and all associated files (including the Inventory Test application) in a folder titled as in the following example “RBurke_Hwk4”. (Use your first initial and last name, of course.) Put this folder in the course Drop Box on Doctor (\Doctor\Assignments\Burke\isds558\HWK4). If you make an error and want to resubmit, that is OK, but you won’t be able to delete, modify or overwrite any folders you have already submitted. Submit a second time, adding a letter to the end of the folder name (e.g. “RBurke_Hwk4A”). I will grade the most recent version submitted (up to class time).

Assessment
This assignment will be assessed on the completeness of the solution to the problem, on use of object-oriented techniques and on the quality of the coding style. Partial solutions will be given partial credit.

Hints and Notes:
- Use the Friend keyword for properties and methods that need to be visible inside the component but not out of it.
- You will need to add your inventory component to the “References” of the application before its classes will be visible.
- DO NOT implement your classes as part of the Inventory Test project. If you do this, you will not be building an ActiveX component. You must create a separate project of type ActiveX DLL.
- The OutOfStock event has an argument that is the name of the item out of stock.
- There can be no negative inventories. You can decide whether to ship everything available or to ignore orders for greater than the stock level.