Online Marketplace System

**Product:**

This class contains the blueprint for any given product on the marketplace. It should keep track of all the following values, which are set initially via the class constructor- there should also be “getter” and “setter” methods (except for Seller ID, which cannot be changed) for each one.

* ID Number (int)
* Name (String)
* Description (String)
* Seller ID Number (int)
* Quantity (int)
* Price (double)

**Person:**

This is an abstract class that contains a blueprint that can be shared by both the Seller and the Buyer. Each of the following values will be set via the constructor- there will be “getters” and “setters” for these values, so that the Seller and Buyer classes can access the private instance variables.

* ID Number (int)
* Name (String)
* Email Address (String)
* Notification (String)

**Admin:**

This interface contains the methods needed to update a given Product in the inventory (for either the Seller Inventory or the Marketplace directly, depending on who is implementing the interface).

* Update quantity of given Product (takes in ID, new quantity)
* Update price of given Product (takes in ID, new price)
* Update name of given Product (takes in ID, new name)
* Update description of given Product (takes in ID, new description)

**Seller (extends Person implements Admin):**

This class inherits all of the methods in the Person class and Admin interface. In the constructor, there will be a “super (…)” call, in addition to the following new variables initialized:

* Seller Inventory (ArrayList<Product>)
* Password (String)

There should also be methods for the following:

* Uploading new Product to Inventory and Marketplace (takes in Product)
* Remove Product from Inventory and Marketplace (takes in ID)
* Displaying all products in Seller Inventory (no parameters)

**Buyer (extends Person):**

This class inherits all of the methods in the Person class. Like, with the Seller class, the constructor will contain a “super (…)” call. There is one new variable:

* Bought Items (ArrayList<Product>)

There should be methods for the following:

* Buying Product (takes in ID)
* Searching for Product in Marketplace (takes in ID)
* Searching for Product in Marketplace (takes in Name)
* Displaying Products in Bought Items (no parameters)

**Marketplace (implements Admin):**

This class contains the blueprint for the entire online market. It should maintain three lists, which are filled by “buyers.csv,” “sellers.csv,” and “inventory.csv” respectively:

* Buyers (ArrayList<Buyer>)
* Sellers (ArrayList<Seller>)
* Inventory (ArrayList<Product>)

There should also be administrative methods for the following:

* Display list of Buyers and Sellers (no parameters)
* Display Buyer purchase history (takes in ID)
* Display Inventory by Seller (takes in ID)
* Add/Remove Buyers and Sellers (takes in Person and ID respectively)- make use of polymorphism
* Update account details for Buyers and Sellers (takes in Person and new info)- Again making use of polymorphism

There should also be two classes related to the actual GUI:

* **Interface**, which includes the frame and panel that the user interacts with
* **Run**, which includes the main method that actually runs the program

*See the UML diagram and code skeletons for a more in-depth view.*