CISC 1110 Assignment 8 – Classes (must be submitted before final exam)

You have recently started a small business that sells smart phones online.

As a startup, you offer a choice of 10 different phones. You hope to eventually grow. However, for now you would like to store the information about your products in a simple database. In this assignment you will be using an array of objects in C++ to implement this database.

The database will have some very basic functionality such as: add, display a phone, print database. (Note: I did not ask you to delete phones from the database.)

Each Phone object must have the following data members:

productnumber: type int -- stores the unique identifying number of each phone

price: type double – stores the retail price of each phone

quantity on-hand: type int -- holds the number of such phones that you have available for sale

name: The name of the phone should itself be an object that stores the company and the name of the phone (both of type string).

Your program will instantiate an array of Phone objects to store your database.

The program will begin by reading in the data from a file. Call a function **readdata** sending it two parameters: the array of objects and n. The function will read in the information about n objects. For each item, read in a value for each of the members of the class. The data must be entered in the input file in the exact order of the members of the class. For the strings, you may use the underscore instead of spaces so that you can read with cin >>. (otherwise, you can use getline() but you will have to use cin.ignore().)

Next, the program will print all the data by calling a function **printdb,** sending it two parameters, the array of objects and n, the number of filled positions in the array. The function will print info for each item in your database in a neat fashion. Note that this function can simply loop and call the function **printPhone** which will print a single Phone object.

Next, the main program will read in a value from the keyboard and call a search function to find that item in the array of objects. You only need to write one search function, but you may write more if you want to. Choose an appropriate name, such as **findprice, findproductnum, findcompany, etc.**

The **findxxx f**unction will receive three parameters: the array of objects, n (number of filled positions in the array), and the value to search for. The search function will return the location of the array that the item is found in. If the item is not found, return -1.

Back in main, the function **printPhone** should be called to print all of the information for the object in the array that matches the search value. For example, if I have the function **findproductnum,** and I search for 1234, I would call **printPhone** to print the productnumber, price, quantity, and name for this phone. The only parameter sent to **printPhone** is the single element in the array (i.e. one object of type Phone), for example mydb[0]. If the search item is not found in the array, main should print an appropriate message.

OPTIONALS: 1. allow the user to enter a range of prices and print all phones within that range. 2. Write a function to **sort** the array by productidnum.