University of Technology, Jamaica
School of Computing and Information Technology

OOP Using Java Lab – Files

a) Create a class called **StockItem** that contains three attributes as follows:

PartNumber - Integer
Description - Character string
Price - Double-precision floating point number

Write a primary and a default constructor for the class.

b) Write a method called DisplayStockItem() in the StockItem class. The method should return no value, and should display the values of the attributes in the class on the screen. A sample display is shown below:

Part Number: 2000
Description: BlackJewelCase
Price: 37.84

c) Write a method called GetStockItemFromUser() in the StockItem class that prompts the user to enter values for all the attributes in the class. The method should return no value. For example, the method should display “Enter part number of item:” then allow the user to enter an integer representing the part number.

d) Write a method in the StockItem class called SaveStockItem(), which stores the current values of attributes in the class to a sequential access text file called “StockItems.txt”. The values of the attributes should remain unchanged after the data is stored to the file and the method should not return a value. The data should be appended to the end of the file if it already exists, and the file should be created if it doesn't currently exist. Use exception handling to appropriately deal with cases where the information could not be stored.

e) Write a method called in the StockItem class called RetrieveStockItem() which prompts the user to enter the stock item number to search for. The method should then browse through the sequential access text file called “StockItems.txt” to see if the stock item number entered matches a PartNumber on file. If it does, it should set the attributes in the class to the information in the corresponding record for that stock item – i.e. the PartNumber, Description and Price attributes should be set to the information in the record retrieved from the file. If the stock item number was not found, then the attributes should be set to some appropriate initial value (e.g. zeros, blanks or empty strings) and the message “Item was not found” displayed on the screen.

f) Write a method called DisplayAllItems() in the StockItem class, that opens the sequential access text file called “StockItems.txt” and displays all the records in it. The information should be displayed in tabular format on the screen, as shown below. No value should be returned from this method and it should not change the value of any of the attributes in the class.

Part Number	Description	Price
1000	GreenTrinketBox	25.44
2000	BlackJewelCase	37.84
2300	PealJewelCase	37.84
3000	RedPerfumeBox	43.21

g) Write a method called RetrievePrice() in the StockItem() class, which prompts the user to enter a stock item number and searches through the sequential access text file called “StockItems.txt” to see if the stock item number entered by the user matches a PartNumber on file. If it does, the method should return the price of the item, but should not change any of the attributes in the class. If the stock item number was not found, an appropriate message should be displayed and the method should return the value -1.0, but should not change any of the attributes in the class.

h) Write a method called RetrieveDescription() in the StockItem() class, which prompts the user to enter a stock item number and searches through the sequential access text file called “StockItems.txt” to see if the stock item number entered by the user matches a PartNumber on file. If it does, the method should return the description of the item, but should not change any of the attributes in the class. If the stock item number was not found, an appropriate message should be displayed and the method should return the value -1.0, but should not change any of the attributes in the class.

i) Modify the GetStockItemFromUser() method you wrote in c) above, to use exception handling to prevent your code from crashing when invalid data is entered for the attributes (e.g. “hello” is entered for the PartNumber).

j) Write a driver file to test the functionality of the **StockItem** class. It should declare an object of the class, and invoke all the methods of the class.

k) Modify your driver file to use a loop to allow a user to enter ten (10) items in the “StockItems.txt” file, by calling the appropriate method from the class.

l) Modify your driver file to use an appropriate method from the class to display all records in the “StockItems.txt” file.