CSE214

HOMEWORK - SUMMER 2016

HOMEWORK 2 - due Tuesday, July 26th no later than 7:00PM

REMINDERS:

- **Be sure your code follows the coding style for CSE214.**
- **Make sure you read the warnings about academic dishonesty. *Remember, all work you submit for homework or exams MUST be your own work.***
- **Click here to read through the steps to make sure you hand in your homework successfully. Also, login to your grading account and click "Submit Assignment" to upload and submit your assignment. Make sure that you submit the same files on both systems.**
- **You are NOT allowed to use ArrayList, LinkedList, Vector or any other Java API Data Structure classes to implement this assignment.**
- **You may use Scanner, InputStreamReader, or any other class that you wish for keyboard input.**

Radio-frequency identification (RFID)--the emerging technology that involves small application-specific integrated chips affixed to individual products or pallets that emits a unique identification number over a radio frequency-- promises to revolutionize supply chains in a new era of efficiency, cost savings, and business intelligence. RFID has been around in some form, such as anti-theft immobilizers, for more than a decade. But large organizations, including department stores, are accelerating its adoption by mandating that their suppliers and members use the technology. RFID technology is an automatic way to collect product, place, time or transaction data quickly and easily without human intervention. It comprises a wireless reader (or interrogator), its associated antenna and small electronic transponders (Tags/ RFID Cards), and related infrastructure that carry the data. The goal of this assignment is to write a Java program that manages a department store's inventory using RFID as a key in a doubly-linked list and also to write a simulator that allows merchandise to be manipulated within the store.

1. Write a fully-documented class named ItemInfo that contains various information about a specific item that can or has been sold in a given department store. Include the product's name(String) and price(a positive double) as fields along with the information in the following format:

-rfidTagNumber: a String that encodes the radio frequency for scanning the item. It is 9 characters long and represented in hexadecimal format(base 16) which means each character is either a digit from 0 to 9 or one of the letters A through F, where case is not important. The length of this String is to be fixed at 9.

-Original Location: a String that encodes the original shelf position of the item. The first

character is 's' to designate that it is a shelf position and it is followed by a 5 digit shelf number to further specify where it can be found in the store. (Examples may be "s00013", "s90909", "s32760", etc.). The length of the String is to be fixed at 6.

-Current Location: a String that represents the location of the item at the current time. It may be a shelf position (as described above), an encoding of a cart number, which is designated by the letter 'c' followed by a 3 digit number ("c101", "c001", "c347", etc.), or it may have been checked out by a customer already in which case the location will be represented by the String "out", where case is not important.

For this class, provide a constructor and accessor/mutator methods for each field. When you set the original position of an item you should also set the current location of that item to the same place. Make sure that you error check the format of each field as specified above and use exceptions to handle this where it is necessary. Write the complete specification for this class as well. ***(NOTE: Because error checking and validation of data is only worth 7 points, DO NOT spend all of your time working on this part of the assignment. A better approach would be to assume, at first, that all input is correct and once you have the rest of the program functional you may go back and validate the data.)***

2. Write a fully-documented class named ItemInfoNode that contains a reference to an ItemInfo object as well as to two other ItemInfoNode objects, referred to as prev and next. Below is a partial specification and it is up to you to fill in the remaining details:

public class ItemInfoNode

- **Constructor for ItemInfoNode**
  public ItemInfoNode()
- **setInfo**
  public void setInfo(ItemInfo info)
- **getInfo**
  public ItemInfo getInfo()
- **setNext**
  public void setNext(ItemInfoNode node)
- **setPrev**
  public void setPrev(ItemInfoNode node)
- **getNext**
  public ItemInfoNode getNext()
- **getPrev**
  public ItemInfoNode getPrev()

3. Write a fully-documented class named ItemList that contains references to the head and tail of a list of ItemInfoNode nodes. Your class will follow this specification, but you have to fill in the details:

public class ItemList

- **Constructor for ItemList**
  public ItemList()
- **insertInfo**
  public void insertInfo(String name, String rfidTag, double price, String initPosition)

Inserts the info into the list in its correct position based on its rfidTagNumber.
NOTE: Multiple instances of a given item ARE allowed in the list. When you are inserting a duplicate item into the list, the order of items of the same type doesn't matter as long as they are together and, as a group, they appear in the correct spot in the list.
HINT: The list should always be sorted by rfidTagNumber. However, this doesn't mean you run a sorting algorithm on the list. As you insert each info into the sorted list, traverse the list to figure out where the new info should go and insert there. Then the new list is still sorted without running a sorting algorithm.

- **removeAllPurchased**
  public void removeAllPurchased()
  Removes all nodes in the list that have current location listed as "out" and displays a list of all items that have been removed in this fashion.
  NOTE: Do not destroy the list as you do this.
  HINT: You might want to consider writing a helper method that removes a specific node from the list, given a reference to that node, and call it as many times as are needed to complete this process.
- **moveItem**
  public boolean moveItem(String rfidTag, String source, String dest)
  Moves an item with a given rfidTagNumber from a source location to a dest location. The return value indicates whether or not an item of the given rfidTagNumber was found at the given source location. Throw an exception if dest is of an invalid format (that it is not a cart, shelf, or "out") and also if source is equal to "out".
- **printAll**
  public void printAll()
  Prints a neatly formatted list of all items currently in the list. The table should include each item's name, rfidTagNumber, original location, current location, and price. The list must be sorted in order of rfidTagNumber, although duplicate rfidTagNumber entries may be printed in any order.
  NOTE: Do not destroy the list as you do this.
- **printByLocation**
  public void printByLocation(String location)
  Prints a neatly formatted list of all items in a specified current location. The table should include each item's name, rfidTagNumber, original location, current location, and price. The list must be sorted in order of rfidTagNumber, although duplicate rfidTagNumber entries may be printed in any order.
  NOTE: Do not destroy the list as you do this.
- **cleanStore**
  public void cleanStore()
  Take every item that is currently in the store and on the wrong shelf and places it back where it belongs (its original location). Items that are "out" or currently in a cart are not affected by this command. Display a table for all out of place items moved in this fashion, including each item's name, rfidTagNumber, current location(before the move), original location and price.
  NOTE: Do not destroy the list as you do this.
- **checkOut**
  public double checkOut(String cartNumber)
  Goes through a given cart and checks out each item (changes its location to "out"). A neatly formatted list of the items checked out is to be printed and it must be sorted in order of rfidTagNumber, although duplicate rfidTagNumber entries may be printed in any order.

The return value is the total cost for the items that were in the cart. Throw appropriate exceptions for invalid cart numbers.
NOTE: Do not destroy the list as you do this.

SPECIAL NOTE: For these seven methods (not the constructor), you should determine their order of complexity using Big-O notation, where N is the number of infos in the items list. You should write a comment in each method that gives the order of complexity along with a concise but accurate explanation.

4. Write a fully-documented class named DepartmentStore. This class will contain a main method that provides a menu with the following options to interact with the program and update the store inventory information:

- **I (Insert an item into the list)**
  Prompts the user for the information about a new item. (see INPUT FORMAT below).
  If all the data entered by the user is valid, the entry is to be inserted into the list into the correct position.
  If any part of the input is invalid you should report the error to the user and ignore the entry.
- **M (Move an item in the store)**
  Prompts the user for an rfidTagNumber, a source location, and a destination location of an item to move.
  If the item is not found, report the situation to the user and do nothing.
  If there are duplicate items with the same rfidTagNumber in one location, moving any given one of them is fine.
  Because we are assuming that there will be no returns or shoplifting, "out" will not be considered a valid entry for either the source or destination through this command. If it is entered, simply inform the user and ignore the request.
- **L (List by location)**
  Prompts the user for a location in the store and prints the list of all items that are currently in that location, sorted by rfidTagNumber.
  If the location entered is not a valid location, print an error message indicating that this is the case.
  NOTE: DO NOT use a sorting algorithm to sort this list. Also, if there are duplicate items by rfidTagNumber, the order of printing DOES NOT matter.
- **P (Print all items in store)**
  Prints the list of all items that are currently stored in the system, sorted by rfidTagNumber. This includes everything that is on a shelf, in a cart, or "out".
  NOTE: DO NOT use a sorting algorithm to sort this list. Also, if there are duplicate items by rfidTagNumber, the order of printing DOES NOT matter.
- **O (Checkout)**
  Takes a cart number and changes the location of every item in the corresponding cart to "out".
  Print a formatted table of all items that have been checked out in this process.
  Print a message to the user of the total cost of all the items that were in the given cart.
- **C (Clean store)**
  Goes through all items that are currently on a shelf that isn't their original location and

places them back to their original position.
This does not affect items in carts or items that are "out".
Print a formatted table of all items that have been moved in this process.

- **U (Update inventory system)**
  Goes through the store's inventory and removes all items that are listed as being "out" from the system.
  Items that are in carts or on shelves are not affected by this command.
  Print a formatted table of all items that have been removed in this process.
- **R (Print by RFID tag number) (Optional)**
  Goes through the store's inventory and prints all items that have the given rfidTagNumber.
  The table is to be printed in the same format as the other print commands.
- **Q (Exit the program.)**

5. Supply any exception handling class(es) that you need in addition to the classes above.

INPUT FORMAT

- **Strings are not case sensitive.**
- **Menu options are not case sensitive and must be checked for errors.**
- **The length of the name field has a maximum of 20 characters.**
- **You may assume that the entries are of the correct Java type (String, String, String, double), but you must check to see if they themselves are valid for that given field.**
- **All lists must be printed in a nice and tabular form as shown in the sample output. You may use C style formatting as shown in the following example. The example below shows two different ways of displaying the name and address at pre-specified positions 21, 26, 19, and 6 spaces wide. If the '-' flag is given, then it will be left-justified (padding will be on the right), else the region is right-justified. The 's' identifier is for strings, the 'd' identifier is for integers. Giving the additional '0' flag pads an integer with additional zeroes in front.**

```
String name = "Doe Jane";
String address = "32 Bayview Dr.";
String city = "Fishers Island, NY";
int zip = 6390;

System.out.println(String.format("%-21s%-26s%19s%06d", name, address, city, zip));
System.out.printf("%-21s%-26s%19s%06d", name, address, city, zip);

Doe Jane             32 Bayview Dr.             Fishers Island, NY 06390
Doe Jane             32 Bayview Dr.             Fishers Island, NY 06390
```

OUTPUT FORMAT

- **Be sure your prompts and error messages for the user are clear and understandable.**
- **All prices must be printed to exactly two decimal places.**

SAMPLE INPUT/OUPUT

Note: User input is in black, computer generated output appears in blue and comments are in green

```
    Welcome!

    C - Clean store
    I - Insert an item into the list
    L - List by location
    M - Move an item in the store
    O - Checkout
    P - Print all items in store
    R - Print by RFID tag number         (optional - extra credit)
    U - Update inventory system
    Q - Exit the program.


Please select an option: I
Enter the name: Dress Shirt
Enter the RFID: 00A5532FF
Enter the original location: s12345
Enter the price: 30.00


//Main Menu is printed

Please select an option: I
Enter the name: Red Towel
Enter the RFID: 0F999FABC
Enter the original location: s00347
Enter the price: 18.00

//Main Menu is printed

Please select an option: I
Enter the name: Silverware Set
Enter the RFID: A1111DDFF
Enter the original location: s90210
Enter the price: 50.00

//Main Menu is printed

Please select an option: I
Enter the name: Dress Shirt
Enter the RFID: 00A5532FF
Enter the original location: s12345
Enter the price: 30.00

//Main Menu is printed

Please select an option: I
Enter the name: Blue Towel
Enter the RFID: 0F999FCBA
Enter the original location: s00347
Enter the price: 18.00


//Main Menu is printed

Please select an option: I
Enter the name: Dress Shirt
Enter the RFID: 00A5532FF
Enter the original location: s12345
Enter the price: 30.00


Please select an option: P


                                 Original        Current
Item Name          RFID          Location        Location     Price
---------        ---------      ---------        ---------   ------
Dress Shirt      00A5532FF        s12345          s12345      30.00
Dress Shirt      00A5532FF        s12345          s12345      30.00
Dress Shirt      00A5532FF        s12345          s12345      30.00
Red Towel        0F999FABC        s00347          s00347      18.00
Blue Towel       0F999FCBA        s00347          s00347      18.00
Silverware Set   A1111DDFF        s90210          s90210      50.00


//Main Menu is printed

Please select an option: M
Enter the RFID: 00A5532FF
Enter the original location: s12345
Enter the new location: c105


//Main Menu is printed

Please select an option: M
Enter the RFID: 00A5532FF
Enter the original location: s12345
Enter the new location: c109


//Main Menu is printed

Please select an option: M
Enter the RFID: 00A5532FF
Enter the original location: c109
Enter the new location: s10000


//Main Menu is printed

Please select an option: M
Enter the RFID: A1111DDFF
Enter the original location: s90210
Enter the new location: s12345


Please select an option: P


                                 Original        Current
Item Name          RFID          Location        Location     Price
---------        ---------      ---------        ---------   ------
Dress Shirt      00A5532FF        s12345          s12345      30.00
Dress Shirt      00A5532FF        s12345          s10000      30.00
Dress Shirt     00A5532FF       s12345          c105        30.00
Red Towel       0F999FABC       s00347          s00347      18.00
Blue Towel      0F999FCBA       s00347          s00347      18.00
Silverware Set  A1111DDFF       s90210          s12345      50.00


//Main Menu is printed

Please select an option: O
Enter the cart number: c105

                               Original        Current
Item Name         RFID         Location        Location     Price
---------       ---------     ---------       ---------    ------
Dress Shirt     00A5532FF       s12345          c105        30.00

The total cost for all merchandise in cart 105 was $30.00

//Main Menu is printed

Please select an option: P


                               Original        Current
Item Name         RFID         Location        Location     Price
---------       ---------     ---------       ---------    ------
Dress Shirt     00A5532FF       s12345          s12345      30.00
Dress Shirt     00A5532FF       s12345          s10000      30.00
Dress Shirt     00A5532FF       s12345          out         30.00
Red Towel       0F999FABC       s00347          s00347      18.00
Blue Towel      0F999FCBA       s00347          s00347      18.00
Silverware Set  A1111DDFF       s90210          s12345      50.00


//Main Menu is printed

Please select an option: L
Enter the location: s12345


                               Original        Current
Item Name         RFID         Location        Location     Price
---------       ---------     ---------       ---------    ------
Dress Shirt     00A5532FF       s12345          s12345      30.00
Silverware Set  A1111DDFF       s90210          s12345      50.00



//Main Menu is printed

Please select an option: C

The following item(s) have been moved back to their original locations:

                               Original        Current
Item Name         RFID         Location        Location     Price
---------       ---------     ---------       ---------    ------
Dress Shirt     00A5532FF       s12345         s10000       30.00
Silverware Set  A1111DDFF       s90210         s12345       50.00
```

```
//Main Menu is printed

Please select an option: U

The following item(s) have removed from the system:

                                Original         Current
Item Name          RFID         Location         Location      Price
---------       ---------      ---------        ---------     ------
Dress Shirt     00A5532FF       s12345            out          30.00


//Main Menu is printed

Please select an option: P


                                Original         Current
Item Name          RFID         Location         Location      Price
---------       ---------      ---------        ---------     ------
Dress Shirt     00A5532FF       s12345           s12345        30.00
Dress Shirt     00A5532FF       s12345           s12345        30.00
Red Towel       0F999FABC       s00347           s00347        18.00
Blue Towel      0F999FCBA       s00347           s00347        18.00
Silverware Set  A1111DDFF       s90210           s90210        50.00

//Main Menu is printed

Please select an option: Q

Goodbye!
```
