

EECS 1510 Object Oriented Programming

Project 6 – Inventory Management

120 Points Due in class Thursday April 13, 2017

Note: This project and Project 7 may be done in pairs, **but the partners must be different on each project; if not, a 60 point deduction.** Also, any code longer than 5 or 6 lines that is identical to another project is grounds for cheating and **an automatic 60 point deduction or even a grade of 0 on the project.**

Consider a program to enter codes of one to eight characters along with an associated item number and associated notes. A code can represent an item, package, or product's name.

By using such a program, we could enter product quantities and additional notes (i.e. storage considerations) before associating them with some brief code (i.e. string), which would indicate the product's ID. We could then retrieve a product's information by entering the product's ID. This might be a useful application for a small store's inventory and stock management.

*Codes are entered as 1 to 8 characters.
Use "e" for enter, "f" for find, "l" for list, "q" for quit.*

*Command: e Soda
Enter quantity: 20
Enter notes:*

*Command: e Milk
Enter quantity: 10
Enter notes: Chilled storage*

*Command: e Chips
Enter quantity: 25
Enter notes: Easily crushed*

*Command: f Milk
-- Milk
-- 10
-- Chilled storage*

*Command: f Popcorn
** No entry with code Popcorn*

Command: . . .

There is also an "l" command, which will list all entries on the screen.

For the input to this program, upper and lower case letters are considered equivalent. For example, if a product name with the code **"Soda" is entered**, then the codes **"SODA"**, **"soda"**, and **"SOda"** will all retrieve the same entry with **"Soda" associated**.

The entries are to be stored in a file from run to run. When the program begins, the entries in the file are to be read into an array. The array should allow for up to 200 entries. The inventory need not be kept in alphabetical order. You may use a simple sequential search to retrieve entries. When the program is exited, the entries should be stored back in a file for use when the program is run again.

Required Program Characteristics: The assignment is to write a program that incorporates the above features. The entries in the inventory manager are to be represented with a class:

```
class Entry {  
    public String name, number, notes;  
}
```

Use an array to store the entries. The array should allow for up to 200 entries.

```
public Entry[] entryList = new Entry[200];
```

Use a static method to read the entries into the program from a file, and a static method to store the entries back into the file when the program is exited.

```
public static void readInventory (String FileName)  
    throws Exception  
public static void storeInventory (String FileName)  
    throws Exception
```

Also, use a static method to list all entries:

```
public static void listAllEntries()
```

Final Submission: See the posted file **"Project Submissions.doc"**.

- In the printouts of the sample runs, each of the commands "e", "f", "l" and "q" should be illustrated.
- You must also show that the external file is updated with entries added from a previous run.

Writing to a File

```
public static void WriteInventory(String FileName) throws Exception{  
    PrintStream P = new PrintStream(FileName);  
  
    for (int i=0; i < num_entries; i++) {  
        P.println(entryList[i].name + "\t" +  
            entryList[i].quantity + "\t" +  
            entryList[i].notes);  
    }  
    P.close();  
    System.out.println("Inventory stored.")  
}
```