CS3354 – Fall 2017 – Assignment 1

Due date: Wed Sep 20 at 11:55 p.m.

Project Title: Shipping Store (stage 1)

Goal: The goal of this assignment is to help students familiarize themselves with the following Java programming concepts:

1. Input/Output to and from the terminal.
2. Storing data in a file and reading data from a file.
3. Creating object-oriented classes and methods to handle data.
4. Using data structures to store data in main memory (e.g. ArrayList).
5. Working with character strings.
6. Using Javadoc comments and generating and html API of the program.

Description:

For this assignment you will create a program that simulates a shipping store database. The database will maintain records of the packages that are in the storage room. Your program should provide the user (shipping truck operator) with a command line choice menu about possible actions that they can perform. The choices should be the following:

1. Show all existing package records in the database (in any order).
2. Add new package record to the database.
3. Delete package record from a database.
4. Search for a package (given its tracking number).
5. Show a list of packages within a given weight range.
6. Exit program.

To represent a package in your program, create a class named Package with the following fields:

- Tracking number (string of length 5)
- Type (string)
- Specification (string)
- Mailing Class (string)
- Weight (floating point number) – in oz
- Volume (integer number) – in cubic inches, calculated as Width x Length x Height

Type: package can be one of the following types: Postcard, Letter, Envelope, Packet, Box, Crate, Drum, Roll, Tube.

Specification: Fragile, Books, Catalogs, Do-not-Bend, N/A – one per package.

Mailing class: First-Class, Priority, Retail, Ground, Metro.

When the program first loads, it reads all the saved records (if any) from a file named packages.txt into an ArrayList. While the program is running, the user can choose any of the 6 available options. When the user selects the option 6 (exit program), the program stores the

current contents of the ArrayList to the file (replacing the old ones) and exits. During the program execution, if the user chooses to add or delete a package, only the ArrayList will be updated. The packages.txt file will be updated only when the program is about to exit. In other words, when the users selects option 6, the program first saves all the contents of the ArrayList into the text file, and then exits.

The format of the contents of the packages.txt file should be in human readable plain text, one record per line. For example:

```
632VR Letter n/a First 1.55 6
H43SM Envelope Do-not-Bend Priority 3.0 12
83RS9 Packet Fragile Ground 55.0 24
GFR23 Box Books Retail 9500.00 45
```

When displaying package records, output headers and make the data line up in columns under the headers.
e.g.

```
-----------------------------------------------------------------------
|TRACKING #|    TYPE | SPECIFICATION |      CLASS |  WEIGHT | VOLUME |
-----------------------------------------------------------------------
|    632VR |  Letter |           n/a |      First |    1.55 |      6 |
|    83RS9 |  Packet |       Fragile |     Ground |   55.00 |     24 |
-----------------------------------------------------------------------
```

NOTES:

- **This assignment can be done individually or with a partner. Let the instructor know who you will be working with ahead of time.**
- You may use an IDE (BlueJ, Netbeans, etc) or just an editor and command line operations (javac, java) in Unix or Windows/DOS to develop your program.
- Use good design (don't put everything in one class).
- Use a package for your classes and put your files in the appropriate directory structure.
- Weight should be output in oz, showing two decimal digits: (e.g 99.95 oz).
- Be sure to validate the user input.
- You don't need to create any GUI for this assignment. Command line operations are enough.
- Use a standard Java coding style to improve your program's visual appearance and make it more readable. I suggest the BlueJ coding style:http://www.bluej.org/objects-first/styleguide.html
- **Use javadoc comments for all of your classes and methods.**

Logistics: Please submit your files in a single zip file (assign1_xxxxxx.zip or assign1_xxxxxx_yyyyyy.zip). The xxxxxx and yyyyyy are your TX State NetIDs.

Submit: an electronic copy only, using the Assignments tool on the TRACS website. Submit using the TRACS account of just ONE member of your partnership. State both names in the comments.
Do NOT include executable .class or .jar files in your submission. Only .java files.