CS3354 – Fall 2017 – Assignment 2

Due date: Tuesday, Oct 3 at 11:55 p.m.

Project Title: Shipping Store (stage 2)

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

- Object Oriented Design
- Inheritance
- Java Exception Handling
- Java Object Serialization

Description:

For the first assignment, you created a program that simulates a shipping store database. The shipping store has now decided to expand its business. It has decided to not just keep an inventory of the packages, but to also record user information and keep a record of the packages user ships with the store. To keep up with the new demands, the shipping store needs to expand its management software to support the new information. For every package, the software will store the same basic information as in assignment 1, plus some additional information:

General information about all packages:

- Tracking number (string of length 5)
- Specification (string)
- Mailing Class (string)

Depending on the specific type of package, the additional information that needs to be in the database is:

Type of Package	Additional Information Needed
Envelope	• Height (integer number) – in inches • Width (integer number) – in inches
Box	• Largest dimension (integer number) – in inches • Volume (integer number) – in inches^3
Crate	• Maximum Load Weight (lb): float • Content: string
Drum	• Material: string (Plastic, Fiber) • Diameter: float – in inches

The database will still maintain records of the packages that are in the storage room. The shipping store now also keeps records of the users. The users are of two types, employees and customers.

- For all user types, the following information is kept in the system:
    - ID (unique number generated for each user): int
    - First Name: String
    - Last Name: String

In addition, the following information is kept for each user type.

- Employee
    - Social Security Number: int
    - Monthly salary: float
    - Direct deposit bank account number: int

- Customer
    - Phone number: String
    - Address: String

Finally, the shipping store records information about completed transactions (delivered packages). Each completed transaction includes the following information:

- Customer id: int
- Package Tracking Number: String
- Shipping date: Date
- Delivery date: Date
- Cost of shipping: float
- Employee id (who completed the sale): int

Note that the customer id, the employee id and the package tracking number included in a sale transaction, should match existing entries of customers, employees and packages already existing in the repository. New functionality of the shipping store system looks like this:

1. Show all existing package records in the database (sorted by tracking number).
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 users in the database.
6. Add new user to the database.
7. Update user info (given their id).
8. Complete a shipping transaction.
9. Show completed shipping transactions.
10. Exit program.

For #2, when the user selects to add a new package to the database, they should be asked to specify the type of the package before they enter the package information. The same applies for #7 regarding the user types.

For #8, when a package is delivered, a new completed transaction record is created and the package is automatically deleted from the package inventory. The program should not allow a transaction to be completed if any of the packages or user does not exist in the database. A user has to be added to the database before a transaction can be completed.

Tasks:

1. Create and object oriented Java program that implements the shipping store management software described above. Make use of the inheritance and polymorphism concepts that we saw in class. For example a print() method used to print information about a package or a user, may be defined multiple times in the different classes and subclasses used in the program.
2. Try to make your program as robust as possible, by using Exception handling to deal with possible problems that may occur during the program execution. You may also create your own custom exception classes to handle bad input from the user, etc.
3. This time do not save your data as text files, **but as serializable objects**. You may use more than one file to store your data, if it is more convenient to you.
4. Print out the package and user information in a formatted manner. Output headers and make the data line up in columns under the headers.
5. 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
6. Use Javadoc is for every public class, and every public or protected member of such a class.
7. Other classes and members still have Javadoc as needed. Whenever an implementation comment would be used to define the overall purpose or behavior of a class, method or field, that comment is written as Javadoc instead. (It's more uniform, and more tool-friendly.)

Notes:

This assignment should be done in pairs of two students. Please find a partner and send the names of your team members to our teaching assistant
Junye Wen <j_w236@txstate.edu> by Tuesday, Sept. 26.

- 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.
- 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_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.