# CS3354 – Fall 2017 – Assignment 5

## Due date: Tuesday, Nov. 21, 2016 at 11:55 pm.

### Goal:

The goal of this assignment is to help students understand the concepts of Java GUI development and Java Logging.

### Description:

This assignment extends the shipping store software from Assignment 2, by replacing console-based user interface with a graphical user interface (GUI), and adding Java Logging functionality. [log errors, log when new package is added, or removed,]

All the operations remain the same as in assignment 2, but this time the user input/output will be handled through a Swing-based Java GUI. The decision regarding which Swing components should be used to achieve the required functionality is left to the students. The students may use the solution of assignment 2 provided by the instructor, but they are free to modify the given code to fit their needs. In the provided solution, the class `MainApp` acts as the main user interface and the class `ShippingStore` acts as the main controller class of the program. After adding the GUI, all the user input/output will be handled by the GUI. Hence, the `MainApp` class will have to be re-written from scratch, whereas, the `ShippingStore` class will not require many changes.

In addition, Java Logging should be used to log the user actions and possible errors or exceptions that may occur during the program execution. Every user interaction with the GUI (e.g. adding a vehicle, showing the list of vehicles, completing a sale transaction, etc.) should be logged at the INFO level and every error or exception should be logged at the WARNING or SEVERE level, depending on its severity. The log output should be saved in a log file, in a human readable (not XML) format. *Hint*: to log user interactions with the GUI, you can add a log statement at each Observer included in your program. Errors can be logged inside the catch part of try-catch statements.

### Tasks:

1. Implement the GUI and Logging functionality without using a WYSIWYG GUI editor. You may still use your favorite IDE to write the code, but you should add all the GUI components manually by directly editing the code.

2. You may reuse code that was provided by the instructor or useful code that you may find from other sources to help you build your GUI (e.g. https://docs.oracle.com/javase/tutorial/uiswing/components/index.html, http://docs.oracle.com/javase/tutorial/uiswing/examples/misc/index.html)

3. 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.

4. 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

5. Use Javadoc is for every public class, and every public or protected member of such a class.
   Other classes and members may 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.)

## Logistics:

This assignment can be done and submitted individually or in groups of two students.
Submit your answer in a single file (assign5_xxxxxx_yyyyyy.zip). The xxxxxx (and yyyyyy) is your TX State NetID.

Submit an electronic copy only, using the Assignments tool on the TRACS website for this class. Do NOT include executable or .class files in your submission.