Course: Advanced Object Oriented Programming
Term: W2017

Assignment 3

Due: Monday April 24 **Weight:** 20% of final mark

Submission:

- Submit all files (the full project package from NetBeans) through blackboard in zipped format (.zip) – **NOTE: I will only accept .zip format – all other formats will not be accepted;**

Requirements:

Solve the requirements listed below using Java programming code only. You must use the concepts we work on in class. All code must be contain full comments. This is a group assignment and all students must contribute in order to receive a mark. You must follow the instructions given below in order to receive full marks. If you require any clarification, please email the instructor. No extensions can or will be given unless approved by the instructor prior to the due date.

Tasks:

Your task is to solve the following programming problems. You must use concepts discussed in class.
Create a database that will hold all the relevant data. The database must be normalized.

The following requirements must be added to the program:

Dashboard window:

This is the entrypoint to the application. It will have a login window to start and once the user has been authenticated, tabs will appear to the following functionalities:

- **Pump management tab – management only:**
  - This should allow setting the price for all gas types – only managers will have access to change this. Any changes must update to the database.
- **Global tank tab – all employees**
  - This should allow the user to see how much gas has been sold and how much is available from each gas type (Regular, plus and supreme). Set it to 1000 liters to start and remove the amount of gas from each transaction. This must be read from the database.
- **Launch Pump – all employees**
  - This will launch the pump GUI in a new window and allow the user to pre-set an amount. There should be a **START**, **STOP** and **COMPLETE SALE** button. The **START** button starts the sale and will update the numbers on the pump gradually. The STOP will stop the transaction. The transaction should stop once the desired amount (previously set) is reached.

Concepts covered in class: Java classes (with constructors, methods and variables), Inheritance, Polymorphism, Composition, Logic, Comments, UML Class Diagrams.

COMPLETE SALE should finish the transaction and update the Global tank as well as write it to the database. Users should not be allowed to pump if the pre-set amount is 0.

Employees can be hard coded into the database. Make sure to have the following entered:

- A part-time employee with
  - ID: 10000
  - Password: 123
  - non-management
- A full-time employee with
  - ID: 10001
  - Password: 456