# ‘COE528 (Fall 2018)

# Project

***Analysis/Design/Implementation***

**General Lab Rules**

All the necessary files of this lab should be in project directory.

All the java files in this lab should have the following package declaration:

package coe528.project;

**Duration: three weeks.**

This project must be done **individually**.

### Problem Description

You will develop a simple Bank Account Application. The application must be a graphical user interface (GUI) based. There are two kinds of users of this application: Manager and Customer. Here we will assume there is only one manager and zero or more customers who uses the application. Every customer has only one bank account. A customer can login, logout, deposit money, withdraw money, get balance, do online purchase(s). The manager can login, logout, add customer, delete customer.

Every customer has three levels: silver, gold, platinum. When a customer has less than 10000 dollars in her account, she is at silver level. When a customer has 10000 dollars or more but less than 20000 dollars in her account, she is at gold level. When a customer has 20000 dollars or more in her account, she is at platinum level. A customer can do an online purchase using the money in her account. The online purchase must be of 50 dollars or more. When a silver customer does an online purchase using her account, she needs to pay a fee of 20 dollars in addition to the purchase amount from her account. When a gold customer does an online purchase using her account, she needs to pay a fee of 10 dollars. When a platinum customer does an online purchase, she does it without paying any fee.

In this application, the manager has username: admin, password:admin, and role = manager. A customer has a username, password, and role = customer, bank account and level. The information about the customers is stored in separate files—one file per customer; the filename can have the username of the customer. When a manager tries to login through the user interface, the manager’s username, password and role are authenticated. When a customer tries to login through the user interface, the customer’s username, password and role are authenticated using the information stored in the relevant file. Only the manager of the bank has the authority to add or delete a customer. When the manager adds a customer, she must create the account of the customer as well with a 100 dollar balance in the account. When the manager deletes a customer, the associated account should get deleted as well. It is assumed that no two users can have the same username.

**The GUI of the application has to be developed using JavaFX.**

### Work Items

1. Develop the UML diagrams for your project using Violet UML Editor (2 kinds – **Use Case** Diagram and **Class** diagram(s).
2. **For** **any one class**, you must:
   1. write the Overview clause stating the responsibility of the class and whether the class is mutable or not. Provide this as javadoc comments.
   2. write the abstraction function and the rep invariant as javadoc comments.
   3. provide the necessary clauses (e.g. effects, modifies and requires) for each method as javadoc comments for the method
   4. implement the abstraction function in the toString() method
   5. implement the rep invariant in the repOk() method
3. Your project must apply the **State design pattern**. The change of level of the customer is the responsibility of the concrete state classes.
   1. A written **report** must be generated. The report should not exceed 3 pages. Use Times New Roman font size 12. You **must** write your **name**, **student ID** and **section number** at the **top of the report**. Your report should **include** the following parts:
      * Describe your Use Case Diagram in a paragraph. Describe one of the use cases following the template given in the file Week.6.1.Modeling.with.UML.pdf.
      * Describe your Class Diagram in a paragraph.
      * Mention the class you have selected to address point number 2 (mentioned above).
      * Refer to your UML class diagram and indicate the part(s) that form the State design pattern.
      * If you use any external sources such as books or papers, you must list them under a section named “References”.
   2. The report will be assessed not only on their technical or academic merit, but also on the communication skills of the authors as exhibited through the report.
4. You **must** give a demonstration of your project to your TA during **your lab session** **on the Project Demo Dates**. The demonstration should include **successful compilation and execution** of your project application.

### Submitting your project

**Project Demo Dates: See announcement in D2L for Project Demo Dates.**

You must include the duly filled and signed standard cover page with your submission. The cover page can be found on the departmental web site:[Standard Assignment/Lab Cover Page](http://www.ee.ryerson.ca/guides/Standard_Cover_Page_Assignments.pdf)

Please ensure that the source code files can be opened, compiled and executed in the lab environment before submission. Please ensure that the Violet UML files and the report can be opened and read in the lab environment.

The **project.zip** file should contain:

* the **report**
* **Violet UML files**
* the **implementation** of the project (e.g. source code)
* Javadoc **for the class** that you have selected to address point number 2 (mentioned earlier).