COE528 (Fall 2016)
Lab5

General Lab Rules
All the necessary files of this lab should be in lab5 directory.
All the java files in this lab should have the following package declaration:

```
package coe528.lab5;
```

Duration: two weeks.

Objectives

- UML Modeling - Analyze system requirements of a software system, design with UML diagrams. You need to use Violet UML Editor tool for drawing the UML diagrams.
  Implement and test the system. You need to use Netbeans for this purpose.

How to start the Violet UML Editor in the lab:

```
Click as follows:
Applications -> Programming -> Violet UML Editor
```

System Requirements

You would need to design a menu-driven, command-line interface based Banking Application of ABC bank. The customers use it to view account balances in their chequing and savings accounts and execute a monetary transaction which is transferring money between chequing and savings account. We assume that a customer has at least a chequing account. No customer can have more than one account of the same type, for example, a customer John Doe cannot have two chequing or two savings accounts. The bank also has a manager who has administrator privileges. In the application, a customer or a manager is represented by a user profile that contains the user name, password and role. The role can be either “Customer” or “Manager”. The active user profile objects are stored in a text file (For creating, reading and writing to files, there is a tutorial: Reading, Writing, and Creating Files you might like to refer to). When a customer tries to login through the command-line interface, the customer’s name and password are authenticated using the information stored in the file. The user profile of the manager (username: admin, password: admin, role: “Manager”) is created in the file during the startup of the application. Only the manager of the bank has the authority to add or delete a customer profile. When the manager adds a profile for a customer, she or he must create the accounts (chequing, savings) of the customer as well with a minimum of 20 dollar balance for an account. When the manager deletes a customer profile, all the associated accounts should get deleted as well. It is assumed that no two users can have the same username.

Follow the process given below to design and develop the required banking application.

Analysis and Design Process

1. Create a Netbeans project named "Ex1" under lab5 directory.
2. Analyze the written requirements.
    a. Identify the *actors*. List the humans or group(s) that interacts with the system under design. Also list any systems that interact with the system under design.
    b. Identify the *use cases*. List the functionality or services provided.
3. Create a UML Use Case Diagram that shows the actors and the use cases. Draw the associations and the relationships between the use cases.
4. Identify at least 4 classes and their attributes and methods.
5. Create a UML Class Diagram containing those classes and show the relationship among the classes where relevant.
6. Create a UML State diagram for one of the classes you have identified above. The state diagram should contain at least two states.
7. Draw the sequence diagram for any two use cases.

All the UML diagrams need to be saved under lab5 directory.

Please note: A UML diagram might need to be revisited if required to fine-tune the information. For example, if you have finished the class diagram and currently doing a sequence diagram, it is possible that you would need to consider a new message in the sequence diagram and therefore you might have to go back to the class diagram to incorporate the method corresponding to the message.

Implementation and Test

You would need to do the implementation and testing in Netbeans.

1. Add all the classes in the Netbeans project.
2. Using the previously developed UML diagrams, update/create attributes and methods in the classes and fill up the method bodies.
3. Compile the Ex1 project.
4. To run your application you need to have the **main** method in the class that is responsible for interacting with the user, i.e. presenting appropriate menu. Test/Run your code with the following instructions:
    - Login to the system as manager with username: admin, password: admin
    - Add a user profile with username: joe, Password: tim, role: "Customer" with chequing account balance equals 500 dollars and saving account balance equals 100 dollars.
    - Add another user profile with username: sue, Password: dim, role: "Customer" with chequing account balance equals 300 dollars and no saving account.
    - Logout as manager
    - Login as user sue
    - sue enquires balance of chequeing account (300 dollars should be displayed back)
    - Logout as sue.
    - Login with username john. (Login should be denied)

- Login as user joe.
- Transfer 100 dollars from chequeing to savings.
- joe enquires balance of chequeing account (400 dollars should be displayed back)
- joe enquires balance of savings account (200 dollars should be displayed back)
- Logout as joe.

Submitting your lab

You must submit your lab electronically at least 24 hours prior to the start of your scheduled lab period for the Project.

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

If you did the lab on a Departmental computer, you can do the following:

```
cd coe528
zip -r lab5.zip lab5
submit coe528 lab5 lab5.zip
```

If you did the lab on your own computer, zip the lab5 folder (remember to do this recursively so that all sub-folders are included), then transfer the zip file to a Departmental machine, logon to a Departmental machine which can be done remotely) and type in the submit command:

```
submit coe528 lab5 lab5.zip
```