BTP400 Lab Activity #1 (Individual Work)

Focus: Fundamental Object-Oriented Programming Skills in Java.

Notes:

1. **The lab is due by 11:58 pm according to the course timeline. No extension will be given. You must submit your Java source file AND a screenshot to Blackboard by 11:58 pm. You will receive a ZERO if you miss the due date. Only one submission is allowed. Internet issues and system problems will not be considered for any extension. Please submit your work as early as possible. Careless mistakes in submission will not be considered as well.**

2. **In order to receive full marks, you must give a DEMO and provide correct answers during CODE REVIEW. Otherwise you may receive 50% (maximum).**

Part A: Create a Java class and testing code.

1. Create a Java class called **Account**. An Account object holds a person's full name (string), an account number (string) and the current account balance (int). You must code the zero-argument constructor, a constructor that takes three arguments, and the toString() method. You must also code three setters and three getters.

   Notes:
   a) If a null value is passed as a parameter, the constructor or method must store an empty string ("") in the object.
   b) If a negative value is passed as a parameter, the constructor or method must store zero in the object.
   b) You should name the setters as setFullName, setAccountNumber and setAccountBalance.
   c) You should name the getters as getFullName, getAccountNumber and getAccountBalance.

2. Create another Java class called **AccountTester** that has only the main method. The main method is used to test that the Account class works properly. Here are the testing requirements.
   a) Create three different Account objects with <u>descriptive</u> data (i.e. real-life data). In particular, one Account object is initialized by a constructor that takes null, null and -1 as parameters. The second Account object should contain your full name.

   b) Write Java code to demonstrate that those three objects have been initialized correctly. Note: Use System.out.println() to display information about an object in the following format.

```
Account Name   : Peter Liu
Account Number: TD12345
Balance            : $1000.99
```

   Note: Be careful with the formatting details.

3. Compile and run your Java programs in a command prompt window, NOT IN ANY IDE.

4. Documentation
   a) You must use **javacdoc comments** to document the class and the methods. You must use at least these four tags: @author, @version, @param and @return. Following the @author tag, you must also include the date that your source code has been created or most recently modified.
   b) You should use C-like comments to document complex programming logic.
   c) You may receive 69% (maximum) if you do not document your code.

5. Submit your work to Blackboard.

Part B (DO IT AT HOME): Understand the Java Platform.

Note: You should do some reading at home. No submission is required. Reading will help you to answer conceptual questions in the tests and final examination.

1. Read The Java Tutorials: About the Java Technology.
   http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html.

2. Answer the following five questions.
a) What does a Java compiler do?

b) Assuming that the Java program does not have any syntax errors.
   What is the name of the file generated by the compiler if the user has entered the following command:

```
>javac HelloWorld.java
```

c) What does a Java Virtual Machine do?

d) Do different operating system platforms (e.g. Windows, Linux, Mac) use identical Java Virtual Machines? Please explain briefly.

c) I have created a Java program (e.g. HelloWorld.java) on the Windows platform. What must I do if I want to run this Java program on the Linux platform?