INHERITANCE AND POLOMORPHISM

1. OBJECTIVE

To review techniques of Java class inheritance and polymorphism.

2. LABORATORY

This lab will be conducted in the Computing Lab 1 (N4-B1-8) in SCE. This is an individual experiment. You cannot use the classes that implement the **Collection** interface (please refer to http://java.sun.com/docs/books/tutorial/collections/), except for Array class. We are going to implement these classes (like Vector, LinkedList, Stack, TreeSet, etc.) in CPE/CSC105, instead of using them directly from Java.

No make-up is allowed for students who have missed their stipulated lab classes without any acceptable excuse like having a valid leave of absence from the school or on medical leave. The students will be deemed to have failed the particular lab work. In case you have valid reasons to do makeup, you must inform the lecturer in charge. The makeup should be done within the same week, during the lab sessions attended by other groups.

3. EQUIPMENT

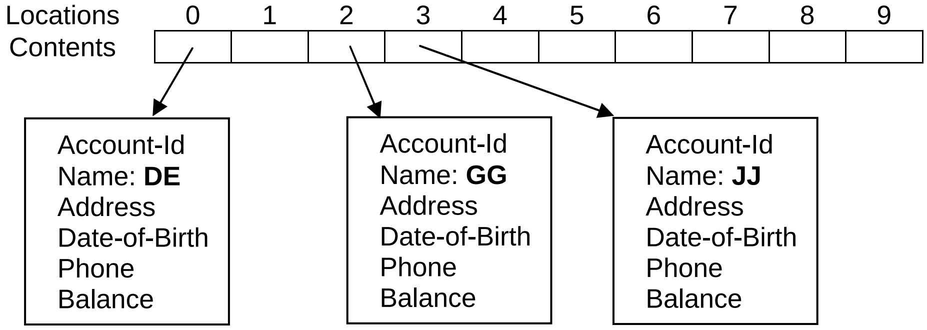
Hardware: The PCs running under the LINUX environment in Computing Lab 1.
Software: NetBeans, SUN JAVA compiler (**javac**) and interpreter (**java**). Your programming will only be tested by the lab markers using javac and java on the lab PCs. None other is accepted.

4. EXPERIMENT

This lab exercise will last for one session and continues from Lab 2.

(a) Re-declare **Customer** as a super class with 3 sub-classes, **Account1**, **Account2** and **Account3**. **Account1** is a saving account that receives ***daily interest***. **Account2** is a checking account that gets no interest. **Account3** is a fixed deposit account with a ***fixed daily interest rate***. One example of the input file is illustrated in Appendix A where a unique fixed interest rate will follow immediately after a fixed account description. Note that the fixed interest rates can be different for different fixed daily interest accounts (see the first and last records). Furthermore, a person can have more than one account id (see the last 2 records) in our banking system.

(b) Write a method **deleteRecord** that removes a customer record from the ordered array of ascending names. When removing an existing record, a void (empty) slot will result. Since the array must be compact at all times, some records must be left-shifted to fill up the void. Suppose we have deleted a record at location 1 and the array before compacting is as follows:



The void at location 1 must be filled by shifting records with names of GG and JJ by one slot to the left so that the array remains compact.

(c) Write a method **update** that computes the balance of each account at end of a month (assuming current month has 30 days). The interest should be computed in compound interest manner, i.e. at the end of each day, compute daily interest according to the money that has been in the account for the last 24 hours. For example, if you have $100 at hour 0, the money in your account will be $100 plus the daily interests computed according to $100 at hour 24. Make use of polymorphism here to have one update method that can work for all different types of accounts.

(d) Write a method **menu** that displays all the methods for test run. The lab markers can select method 1, 2, 3, 4, 5 or Q to input data, display data, output data, delete record, update and quit in any order such that it is possible for the markers to input or output data more than once. For command 4, the program should prompt and read in the account **id** to be deleted. For command 5, the program should prompt for the current interest rate for the saving account. The update should be done once only. Additional request for update should be ignored on the same account. Duplicate records should be ignored and appropriate error statement should be printed. However, the program should still be run as usual.

Although each of the above tasks is to be implemented as a single method, it may be necessary to create additional sub-method(s) to handle portions of the method. This is especially true if the original method is too long or contains more than one functionally related group of statements. When you write a method, remember that this method is to work for all possible inputs. Not on just your test inputs. You must test for all conditions that might possibly arise; print out error messages as needed.

A text file **customers.txt** of customer records has been created in the directory **/home/staff1/csc105/lab4/** to assist the development and testing of your program. Its content and even the filenames will be changed (while still adhering to the same data format) during grading in order to test your programs' robustness. The objective is to ensure that you do not hardcode your program to work exclusively for the given sample customer records. In addition, a text file **commands** will be provided to test your program, i.e. run your program as "**java Lab4 < commands**". Check also the **readme** file for any last minute hints or changes.

Important: From lab 4 onwards, all your programs should comprise a few files:

- Source code files: a few files to define the classes, e.g.
  - *`Lab4.java`*: this contains the main() method
  - *`Customer.java`*: this contains the Customer class definition
- One readme file that gives instructions to compile and run your program.

Create a sub-directory called **cpe105/lab4** or **csc105/lab4** (depending on whether you are taking CPE105 or CSC105) in your home directory. Please note that the grader will only look into this sub-directory to find, compile and test run your program. You should not have irrelevant source files in the directory.

5. ERROR HANDLING

You can assume that the input file format is correct. No error checking except those from lab 2 is needed.

6. REPORT

This lab is the second part of a larger project composed of labs 2, 4 and 5. You do not need to submit report for this lab. Be prepared to submit the final report of lab 5. You can refer to lab 5 for requirements and report format. In case your lab 5 cannot work but lab 4 can work, you can submit lab 4 report for lab 5. However, you will face great penalty. In addition, lab 6 will be based on lab 5 only.

7. ACADEMIC HONESTY AND COLLABORATION

Cooperation is recommended in understanding various concepts and system features. But the actual solution of the assignments, the programming and debugging must be your individual work, except for what you specifically credit to other sources. (Your grade will be based on your own contribution.) For example, copying without attribution any part of someone else's program is plagiarism, even if you modify it and even if the source is a textbook. You can document the credit to other sources at the start of your program code listing. The University takes acts of cheating and plagiarism very seriously: first time violators may fail the coursework component of CPE105 / CSC105. Any wholly (or partly) copied (or being copied) programs will receive zero mark.

8. REFERENCES

[1] Text and reference books for CSC/CPE 105.

[2] http://java.sun.com/docs/index.html; http://java.sun.com/j2se/

Appendix A: customers.txt

```
Account Id = 123
Name = Matt Damon
Address = 465 Ripley Boulevard, Oscar Mansion, Singapore 7666322
DOB = 10-10-1970
Phone Number = 790-3233
Account Balance = 405600.00
Account Type = Fixed
Fixed Daily Interest = 0.05

Account Id = 126
Name = Ben Affleck
Address = 200 Hunting Street, Singapore 784563
DOB = 25-10-1968
Phone Number = 432-4579
Account Balance = 530045.00
Account Type = Saving

Account Id = 65
Name = Salma Hayek
Address = 45 Mexican Boulevard, Hotel California, Singapore 467822
DOB = 06-04-73
Phone Number = 790-0000
Account Balance = 2345.00
Account Type = Checking

Account Id = 78
Name = Phua Chu Kang
Address = 50 PCK Avenue, Singapore 639798
DOB = 11-08-64
Phone Number = 345-6780
Account Balance = 0.00
Account Type = Checking

Account Id = 234
Name = Zoe Tay
Address = 100 Blue Eyed St, Singapore 456872
DOB = 15-02-68
Phone Number = 456-1234
Account Balance = 600.00
Account Type = Saving

Account Id = 2350
Name = Zoe Tay
Address = 100 Blue Eyed St, Singapore 456872
DOB = 15-02-68
Phone Number = 456-1234
Account Balance = 600.00
Account Type = Fixed
Fixed Daily Interest = 0.055
```