Software Engineering Design – Advanced Programming Techniques, Semester 2022A

RMIT UNIVERSITY

LAB ASSESSMENT 1 (25%) – TEST QUESTIONS

Test Duration: 120 mins (+ 15 mins for submission)

NOTE: only do and submit **one source .cpp file for each question 1-2, and don't zip them together.**

Question 1 (10 pts)

Write a C++ program which can work with **command line arguments** below:

- Get three hexadecimal numbers and print out their sum (all must be preceded by "0x").
  ```
  ./a.exe 0x12 0xAC 0xB3
  Sum is: 0x171
  ```

- Write all arguments (except the first one) into a file namely "**data.txt**", each argument into a line, if the first argument is "**-w**".
  ```
  ./a.exe -w GoodMorning. HowAreYouToday 12345
  ```

  File content:

  CppWorkspace > LabAssessment1 > myFile.txt
  ```
  1  GoodMorning.
  2  HowAreYouToday
  3  12345
  ```

- Read from the file, and print out the longest line, if the first argument is "**-r**"
  ```
  ./a.exe -r
  Longest line: HowAreYouToday
  ```

Hint: *when reading from the file, instead of using eof() function to check for end of the file (which may not work), you can check for length of resulting string.*

Question 2 (15 pts)

a) Defines a class namely **EWallet** with two private attributes are **id** (string) and **balance** (double). Write constructor for it to initialize the attributes, and two methods as below:
   - void **deposit**(double num): to add **num** amount of money into the ewallet.
   - bool **withdraw**(double num): to take out **num** amount of money from the ewallet.

   You should have appropriate message printing to screen when those methods are called.

   Create an object by dynamic allocation in main() function and initialize its data using the constructor, then test all functions above. Free up allocated memory after that.

b) Define another class namely **Customer** with two attributes are **name** (string), **pwd** (string), **ew** (EWallet object). Provide constructor for it, and three methods as below:

   - bool **verifyPass**(): ask the user to input a password, and compare it with attribute **pwd**. Return true if match, and false if does not match with appropriate messages on screen.

   - bool **doDeposit**(): call the method **verifyPass**() first to verify password. It success, allow user to input amount of money for depositing and add it into current balance of the customer's ew object (hint: must call **deposit**() method of the ew object)..

   - bool **doWithdraw**(): call the method **verifyPass**() first to verify password. It success, allow user to input amount of money for withdrawing and subtract it from current balance of the customer's ew object (hint: must call **withdraw**() method of ew object).

c) Overload the >> operator so that we can do following operations
   customerA >> customerB : all money of customerA will be transferred to customerB

   customerA++ (postfix): the current balance of customerA will be increased by 1000, but return all previous value.

d) Create an array of four customers using dynamic allocation and initialize values for them. Store all information into a text file (each customer per line, each information is separate by ',' symbol). Find and print out the customer with second largest balance. Free up allocated memory at the end of the program