Final Examination Part B
Programming Tasks (20%)

Download the file *package*.zip* (* can be any number), which contains the files, *Date.h*, *BaseCode.cpp*, *voucherlist.txt* and *ClaimSheet_FinalExam.xls* (different student may get different *voucherlist* files). The files, *Date.h* and *BaseCode.cpp,* contain the code that implements the basic input/output functionalities in C++ to process two types of government issued vouchers – "**Active Voucher**" ($100) and "**Dine Voucher**" ($25). The base code is not written in pure object-oriented style. Import the package into an IDE. If the program is not runnable in your IDE, change it to meet the requirements of your IDE and make the base code runnable with your IDE. The following tasks aim to convert the base code into more OO style and further extend it with more functionalities:

Task 1 (2%): In the main function, two sample vouchers are created. Change the voucherNumber of the Active Voucher into *your student id* and the name on both vouchers from "*John Smith*" to *your name*. Randomly generate 8 capital letters as the voucherCode of the Dine Voucher. Make sure the program can read the file *voucherlist.txt*.

Task 2 (4%) Implement the class, **Voucher**, to contain the common data members of the classes, **ActiveVoucher** and **DineVoucher**, as shown in *BaseCode.cpp*. Make all these data members to be either **private** or **protected**. Check the free functions in *BaseCode.cpp* to see if they can be member functions of this class or create virtual functions for them to let the derived classes override them. Create constructors and/or destructors if necessary.

Task 3 (6%): Change the classes, **ActiveVoucher** and **DineVoucher**, into derived classes of the **Voucher** class, respectively. Move the related free functions in *BaseCode.cpp* to the respective classes. Remove or change parameters of each function if necessary. Some of these functions may override the pure virtual functions of **Voucher** class. Implement constructors and destructors if necessary.

Task 4 (4%): The file *voucherlist.txt* gives a list of voucher requests. Rewrite the class **ProcessVouchers** so that it can read the *voucherlist.txt* file, create an object of either **ActiveVoucher** or **DineVoucher** for each voucher request (one per line) in the *voucherlist.txt* file based on either "*Active*" or "*Dine*" of the request. Store the pointers of the objects into an array or vector of **Voucher.** You may also store the objects into separate arrays or vectors of ActiveVoucher and DineVoucher, respectively. Partial marks may be given if you do not store the objects but output the vouchers into an text file, named "*voucheroutput.txt*", using *printVoucher()* method of **ActiveVoucher** and **DineVoucher**.

Task 5 (4%): Print the vouchers in the array/vector using the *printVoucher()* method of each derived class of Voucher to a text file, named "*voucheroutput.txt*". Further extend your code so that the output vouchers are listed in the file in ascending order of expiry dates of the vouchers. No marks for file output if you have received marks for output from Task 4.

Remarks:
1. You may change the names of files, classes, and functions. You may also add new classes/functions/files as part of your solution.

2. Fill in the *ClaimSheet_FinalExam.xls* with your name, student id and the marks you claim for each task. If some tasks were partially completed, claim partial marks, and put a comment for what you have done in the respective column.
3. Your code must be runnable. If you have tried some of the tasks but did not work, you may comment these parts of code and claim partial marks in your claim spread sheet (no more than half marks of workable implementation).
4. Your submission should include all your source code, *voucherlist.txt*, *voucheroutput.txt* and *ClaimSheet_FinalExam.xls*. Zip the files into a single file with your student id as the file name. Submit the file on vUWS by the end of the examination.