UNIVERSITY OF WOLLONGONG

SCHOOL OF COMPUTER SCIENCE AND SOFTWARE ENGINEERNG

CSCI124 APPLIED PROGRAMMING SPRING 2013

ASSIGNMENT 2 (5 MARKS)

DUE: 11:59PM TUESDAY 10 SEPTEMBER 2013 (WEEK 7)

AIM

Practice of binary file handling and sorting

On completion of the assignment, you should know how to:

- Design a program by decomposing the problem into smaller functional units
- Implement a program in stages
- Produce a C++ database program for maintaining records

REQUIREMENTS

In this task, you are going to enhance the functionality of the database program in assignment 1. Recall that each student record in the database contains the following information.

Name	at most 30 char (cannot contain the newline character, can contain anything else but not NULL)
ID	a unique identifier a positive integer of 7 digits (cannot be NULL)
Course	4 digit course code, a space, then at most 80 char e.g. 1612 Master of Computer Studies (cannot be NULL)
List of Subjects	each subject consists of the following information: 1. year (4 digits) 2. session (one char A/S for autumn/ spring) 3. code (7 characters) 4. credit (2 digits, i.e., if the credit point is 6, it will be represented a 06) 5. Mark (0 to 100 or e or w) e enrolled, w means withdraw You may assume each student can be associated with at most 30 subjects
GPA	the system should be able to calculate the GPA automatically based on credit and mark of the subject list of the student

GPA is defined as:

(Sum of {credit of all subjects multiplied by their scores} divided by Sum of {credits of all subjects}) multiplied by 6 divided by 100

All subjects here means subjects that is not of the status e and w.

New commands in addition to those specified in assignment 1 in the main menu:

I	Import from a binary file. The format of a binary file is up to your decision.
E	Export to a binary file. The format of the binary file is up to your decision. The only requirement is that it has to be compatible with the Import from a binary file function.
C	Validate course information. Check for any mismatch between course code and course name. Display the conflicting records and ask the user to choose one of the two options: - ignore - all records to one of these conflicting names
O	Options. Choose the order of display according to the three sort keys from the three fields: Name, ID and GPA Records should always be displayed according to the sort key and order. Default option is shown below: First Sort Key: ID , ascending Second Sort Key: Name, ascending Third Sort Key: GPA, descending *Remark: it means that records should be ordered according to the first sort key. If they have the same first sort key, they are ordered according to the second sort key (For easy implementation, it is possible for the user to set all three sort key to the same field)* Records should always display according to the setting in this menu.

You can assume the maximum size of the database in this assignment is 100.

All commands are case sensitive. If the user enters an invalid input, your system should prompt the user to re-enter. You can assume the user will never enter anything longer than 100 characters.

GUIDELINES

Do not alter the menu options or input data requirements as they will be used to test your program. Comment your code appropriately.

Implement your program in stages. One possibility is to complete the Export to a binary file function first, followed by the Import from a binary file. In order to test the read function, you may write another “display” function which simply displays all records in memory.

Do not hesitate to ask your lecturer or lab tutor if you are unsure of anything or encounter any difficulties. (Remember it is typically the software developer’s responsibility to clarify the requirements of the software with the client.)

Important: YOU MUST SUBMIT A MAKE FILE that compiles your code on Banshee. Your code should compile without any warnings.

To show all the warnings, you should compile with the -Wall option.

As usual, please start early. I am always happy to discuss with you about the idea.

SUBMISSION

Submit your program (and Makefile, if any) via the submit command.

```
submit –u userid –c CSCI124 –a ass2 filenames
```

Please make sure you receive the submission receipt after submitting your files.

Remember that you have to put the following information on the header of each source file you will be submitting in this assignment:

- Student name
- Student number
- Lab

Please also submit a hard copy of your code to one of your tutors or the lecturer anytime in week 7 (during the lecture, labs, or the consultation time of the lecturer) after you obtained your submission receipt.

An extension of time for the completion of the assignment may be granted in certain circumstances. A request for an extension must be made to the Subject Coordinator before the due date. Supporting documentation must accompany the request for extension. Late assignments without granted extension will be marked but the mark awarded will be reduced by 1 mark for each day late. Assignments more than 3 days late will not be accepted.

For late submission, please use the command:

```
submit –u userid –c CSCI124 –a ass2-late filenames
```