The University of Nottingham

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

A LEVEL 2 MODULE, AUTUMN SEMESTER 2014-2015

SOFTWARE ENGINEERING DESIGN

Time allowed 2 HOURS

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Complete ALL the examination tasks

Please note well: for all tasks marks will be given for the correct declaration of classes and functions even if their definition (implementation) is missing.

<u>You may choose to provide either separate files for each of the tasks or a single solution for the whole exam - just be clear please.</u>

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

DO NOT turn examination paper over until instructed to do so

ADDITIONAL MATERIAL: NONE

Note to Invigilators:

This is an open book examination and candidates may bring in any material on disks, memory sticks etc that they wish.

The course material must be available to the candidates on the computers.

Internet and email access must be disabled on the computers and not available to the candidates.

Introduction:

A lecturer wishes to have a basic software code for recording student marks.

For each student, the software will generate a *student record.*

Each *student record* identifies the student whose marks it contains by means of an integer *student ID number* and also contains a list of *module records.*

Each *module record* contains an integer *module ID number* and an integer *exam mark* (a percentage in the range 0 to 100) which is the student's score in the exam for that module.

Examination Task: part 1 (40%)

Your first task is to design and implement the basic *module record* class. Please provide just the minimum functionality expected of any well designed class. Please make any required design decisions in the context of the project described above and in accordance with good software engineering practice.

Examination Task: part 2 (25%)

Your next task is to design and implement the basic *student record* class. Please provide just the minimum functionality expected of any well designed class. Please make any required design decisions in the context of the project described above and in accordance with good software engineering practice, annotating the code where appropriate.

Examination Task: part 3 (10%)

Overload the stream operators so that the software can write a *student record* to the screen or to a file in the following format.

student_ID_number
N
module_ID_number $_1$ exam_mark $_1$

....

module_ID_number $_N$ exam_mark $_N$

where all the entries shown are integers. N is the number of modules the student took.

Similarly, provide input stream operators to permit reading of *student records* in this format

Examination Task: part 4 (5%)

If the pass *exam mark* for each module is 40%, then add to the class *student record* two member functions, the first returns whether the student has passed all modules and the second returns the number of failed modules.

Examination Task: part 5 (20%)

To provide for future expansion, it is now considered appropriate to generalise the concept of a *module record.* A *module with project record* is to be a sub category of *module record* which, in addition to containing a *module ID number* and an *exam mark*, also contains an integer *project mark* (a percentage in the range 0 to 100).

Please develop the class *module with project record* so that:

1) The code reflects the relationship between *module with project record* and *module record.*

2) The code for tasks 2-4, which only uses basic *module record* objects, does not need to be modified.

3) Referring to task 4: To pass any type of *module* requires an *exam mark* of at least 40%. However, to pass a *module with project* also requires a *project mark of* at least 40%.

End