Beijing-Dublin International College

SEMESTER 2 FINAL EXAMINATION - (2016/2017)

School of Computer Science

COMP2005J Object Oriented Programming Programming Exam

Prof. Pádraig Cunningham
Dr. Seán Russell*

Time Allowed: 180 minutes

Instructions for Candidates:

Answer all questions.

BJUT Student ID:_____ **UCD Student ID:**_____

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

Honesty Pledge:_____ **(Signature)**

Instructions for Invigilators

Students are permitted to bring any printed notes and textbooks into the examination.

- All questions are based on the same program. Each question adds some new functionality to the code. You should not move to the next question until you have finished the one before it.
- You should only submit working code. If your code does not compile or work correctly you should comment it out.
- To improve your grade you should submit well designed code with light comments and correct indentation.
- The main method of the program should be in a class named `Main`.
- There are bonus marks for providing good JUnit test cases for the classes you have written. This is not required to get 100%, but it will improve your score.
- There are files with some input data for the program on moodle. Download these at the beginning of the exam. Internet access will be turned off after 10 minutes. If you have not downloaded the input files you will have to create them yourself. These files should be put in the project folder in eclipse, not in the src folder with your code.
- The classes you define should have good encapsulation. If you need to access an instance variable you should add a getter or setter method.

Question 1: Student Info

Students are required to completed many assignments to pass a module. The results of these assignments must be combined to calculate the grade the student gets. Each student has an id (this can be letters and numbers) and the grades for 5 assignments. Sample data is provided in the **text files** `COMP2005J.txt` and `COMP1001J.txt`. The format of each line in these files is:

```
<id> <a1> <a2> <a3> <a4> <a5>
```

Here `a1-5` represent the result of each of the assignments in order.

a. Implement a `Student` class in Java to represent a single student and the results they achieved in a single module.

(10%)

b. Add a method to the `Student` class to calculate the average result from all 5 assignments for that student.

(10%)

c. When a student completes a module, they are give their grade instead of the result. Write a method to calculate the grade the student should receive based on the following table.

Grade	%
A	>=85
B	>=70
C	>=55
D	>=40
E	>=25
F	>=10
G	>=2
NG	>=0

(10%)

d. Override the `toString` method in the `Student` class, when called this method should print out the students id and their grade.

(10%)

e. Add code to the main method to read all of the student information from the files. A student object should be created for each line and these should be printed to the screen.

(10%)

The output of the main method should look something like this:

```
Student Results in module COMP2005J
ucd1584151 D
ucd1567930 C
...
ucd1592563 C
ucd1516105 D
Student Results in module COMP1001J
ucd1624628 D
ucd1675038 C
...
ucd1694812 A
ucd1605914 C
```

(Total 50%)

Question 2: Module Info

The information about a module is stored in a single file with the information about each student on a single line. However, sometimes it is more useful to know about individual assignments than individual students.

a. Implement a `Module` class in Java to represent all of the results in a single module. The class should represent the code the module is known by and have a method that allows students to be added one at a time.

(10%)

b. Add a method to the `Module` class to find the id of the best student in the module (Highest average result).

(5%)

c. Add a method to the `Module` class to find the id of the worst student in the module (Lowest average result).

(5%)

d. Add a method to the `Module` class to find the average result for a single assignment. This method should take a single int as a parameter to specify which assignment average should be calculated (0 - 4).

(10%)

e. Override the `toString` method in the Module class so that it creates a string containing the code of the module, the id of the best student, the id of the worst student and the average result for each of the assignments.

(10%)

f. Add code to the main method after the code from question 1. This code should again load all of the information from each file, but instead of printing the information about each student, these should be added to the module object. Finally print out the details of each module using the `toString` method.

(10%)

The output of the main method should look something like this:

```
...
Details of module COMP1001J
Best student: ucd1694812
Worst student: ucd1624628
Average result in assignment 1 is 62.1465
Average result in assignment 2 is 63.67749999999999
Average result in assignment 3 is 62.678499999999985
Average result in assignment 4 is 63.349000000000004
Average result in assignment 5 is 63.0655

Details of module COMP2005J
Best student: ucd1571972
Worst student: ucd1518316
Average result in assignment 1 is 52.678
Average result in assignment 2 is 66.09199999999997
Average result in assignment 3 is 54.826499999999996
Average result in assignment 4 is 59.38900000000001
Average result in assignment 5 is 53.855000000000004
```

(Total 50%)

Submission

Create a single **zip** file containing your entire project. This file should be named using your UCD student number.

Add a text file named statement.txt, that contains your name, student ID and a short statement of what you achieved, e.g., "Parts 1-3 completed correctly, Part 4 attempted but unfinished." Call the attention of an invigilator and they will collect your submission.