COM00141M

Department of Computer Science

ALGORITHMS AND DATA STRUCTURES

ASSESSMENT BRIEF

Assessment Author	Tommy Yuan
Assessment	Summative - Open
Release	Week 1
Submission	Monday Week 4 at 13:00 (UK time)
Feedback	Friday Week 6 at 13:00 (UK time)
Weighting	10%

I. Module Learning Outcomes

1. Express a problem solution algorithmically using pseudocode
2. Analyse the time complexity of an algorithm
3. Construct computer programs to implement algorithms
4. Test a computer program against the specification.

This assessment will contribute to learning outcome 3 & 4 for this module.

II. Practical Programming Assignment 1

Introduction

You are required to develop a grading system for the University of York online taught Masters Programmes. The system should enable students to check the grade for a particular module and the degree as a whole.

A Masters degree contains 180 credits with nine taught modules (135 credits: 9 modules and 15 credits each) and two independent study modules (ISMs) (15 + 30=45 credits).

For a particular module, the grading criteria are as follows:

• 70-100	Excellent
• 60-69	Good
• 50-59	Satisfactory
• 40-49	Compensatable fail
• 0-39	Outright fail

For a degree, the grading criteria are as follows:

Distinction: a student must achieve the following at first attempt

i) A rounded credit weighted mean of at least 70 over all modules, and
ii) a rounded credit weighted mean of at least 70 in the Independent Study module(s) taken, and
iii) no failed modules.

Merit: a student must achieve the following at first attempt

i) A rounded credit weighted mean of at least 60 over all modules, and
ii) a rounded credit weighted mean of at least 60 in the ISMs taken, and
iii) no more than 15 credits of Compensatable failed modules, and with no module marks below 40

Pass: a student must achieve the following at the first attempt

i) A rounded credit weighted mean of at least 50 over all modules, and
ii) a rounded credit weighted mean of at least 50 in the ISMs taken, and

iii) no more than 30 credits of Compensatable failed modules, with no module marks below 40

Fail: For students who do not pass on the first attempt there will be an opportunity for a resit if the appropriate conditions are met. Students are entitled to reassessment in a maximum of 30 credits-worth of failed modules provided that they have failed no more than 60 credits with no more than 30 credits-worth of outright fail (i.e. module marks less than 40).

III. Assessment Task/s

Part A

1. Using Eclipse you should create a new project called: York grading project. Within the project, create a ModuleGrader class

2. Within the class, create a method called gradeModule, which takes a mark (e.g. 65) as parameter and returns a grade (e.g. “Good”).

3. Write another method called getValidModuleMark, this method will prompt the user for input from the keyboard and validate a mark in the range 0 to 100. It will keep prompting the user until a valid input has been taken.

4. Write a third method called startModuleGrading, the method will start with displaying "*********** Module Grading Program *********", it will then prompt the user for a module mark to grade. For a valid mark, the grade will be printed on screen. The system will then ask the user whether he/she would like to continue grading , or not, by inputting - yes/no (y/n) - This method needs to call the methods developed in “Part A 2.” and “Part A 3.” above.

5. Write a TestModuleGrader class to test each method.

Part B

1. Within the York grading project, create a DegreeGrader class

2. Within the class, create a method called gradeDegree, which takes four first attempt module results (i.e. all module average, ISM module average, number of compensentable failed credits and number of outright failed modules) as parameters and returns the degree classification (e.g. “distinction”, “merit”, “pass” or “fail”). Here is an example of some of the parameters taken by the gradeDegree method: gradeDegree(44, 55, 14, 2)

3. Write one or more methods that will prompt the user for input from the keyboard for all the module averages: ISM module average, number of compensentable failed credits, and number of outright failed modules. The methods should also validate the input (e.g. module average in the range 0 to 100, number of compensentable failed credits in the range of 0-180, and number of outright failed modules in the range of 0-11). It will keep prompting the user until valid inputs have been entered.

4. Write a further method called startDegreeGrading, the method will start with displaying ("*********** Degree Classification Program *********", it will then prompt for the results to grade. For valid results, the grade will be printed on screen. The system will then ask the user a whether he/she would like to continue grading, or not, by inputting - yes/no (y/n) - This method needs to call the methods developed in “Part B 2.” and “Part B 3.” above.

5. Write a TestDegreeGrader class to test each method.

IV. Deliverables

Please submit the following:

1. A single report in **PDF (or word)** format, which contains evidence of **ALL testing** and outputs (screen shots and snippets of code).
2. A **single zip file** that contains all your java source code files – ModuleGrader, DegreeGrader and the 2 Test classes

V. Marking Criteria

A marking scheme and assessment criteria can be found below:

The marks for each section are detailed below. The maximum mark is 100. This will be scaled to 10% for the assignment overall.

The following contributes to the learning outcomes 3 & 4 as outlined earlier in section I		
Section	**Criteria**	**Available marks**
A. **Class ModuleGrader**	Source code evidence for appropriate attributes, method signatures and data types	**5**
Method gradeModule	Source code evidence for appropriate method signature, data types and method body.	**5**
Method getValidModuleMark	Source code evidence for appropriate method signature, data types and method body.	**5**
Method startGrading	Source code evidence for appropriate method signature and data types and method body.	**5**
Test Class	Source code evidence of the class and Test objects being instantiated and initialised correctly.	**5**
Functionality/Testing	Evidence that all methods have been tested and function correctly (marks scaled accordingly)	**10**

B. **Class DegreeGrader**	Source code evidence for appropriate attributes, method signatures and data types	**5**

Method gradeDegree	Source code evidence for appropriate method signature, data types and method body**.**	**10**
Methods for promoting user input	Source code evidence for appropriate method signature, data types and method body	**10**
Method startGrading	Source code evidence for appropriate method signature and data types and method body	**10**
Test Class	Source code evidence of class and Test objects being instantiated and initialised correctly.	**10**
Functionality /Testing	Evidence that all methods have been tested and function correctly (marks scaled accordingly) via the test object	**10**

General	Source code is nicely presented and self-documenting with comments. Appropriate class, method and attribute naming – follows coding convention with indentation.	**10**
	Total	**100**

NOTE: Non submission of your java source code files will result in 0 (zero) marks for the assignment **as all java files are tested to verify the content in your report.**

VI. Grading

The pass mark for postgraduate modules is 50%. For more information about grades and assessment criteria, please review the 'Assessment and award' section of the York Online Handbook which can be downloaded from the **Orientation Module**.

VII. Assessment submission

You will submit your assessments in the 'Assignments' area of the module in Canvas. Please check your canvas module for the specific submission date for this assignment.

For general assessment guidelines consult your Canvas Module, Orientation Module and for Academic Regulations the University of York's website.

Any queries regarding the details of your assessment should be directed to your module tutor. Any queries regarding assessment procedures should be directed to studentsuccess@online.york.ac.uk.