University of Westminster
School of Electronics and Computer Science

4COSC005/W Software Development 2 – Coursework	
Module leader (IIT)	Dilshard Ahamed
Weighting:	50% of the module
Qualifying mark	30%
Description	Coursework
Learning Outcomes Covered in this Assignment:	LO1, LO3, LO4, LO5.
Handed Out:	8th June, 2024
Due Date	**10th July, 2024**
Expected deliverables	Zip the project directory as **w1234567_sd2_cw.zip** Submit Report and Self Evaluation Form (PDF) Live demo
Method of Submission:	Blackboard
Type of Feedback and Due Date:	Written feedback and marks 15 working days (3 weeks) after the submission deadline. All marks will remain provisional until formally agreed by an Assessment Board.

Assessment regulations
Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission
If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognized that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will dec ide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

Coursework Description

Student Activity Management System

You are to write a management system for handling student results for three modules in Semester 1. The system will print a menu which the university staff will interact to manage students and their module results.

Task 1. **Arrays solution.**

Design a project to manage student's academic results and performance. The project must be capable of handling a university intake. Let's assume the capacity of the intake is 100 so maximum 100 students can be registered and managed. You may can check the sample code given in the Blackboard for your reference. [Sample Code - Task 01]

Once initial program has been planned, create a menu where a university staff can interact. The menu must be included with following options.

1. Check available seats
2. Register student (with ID)
3. Delete student
4. Find student (with student ID)
5. Store student details into a file
6. Load student details from the file to the system
7. View the list of students based on their names **(Implement your own sort algorithm)**

Each of above process to be implemented using methods in JAVA. Example. When user pressed "1" system should call "checkAvailableSeats" method. For "2", "Register" method and so on.

When user selected option "5", all the data that are stored in the array to be transferred to a txt file. When the program stopped and run again user must be able to load the data back to the system again by selecting option "6".

You can build up your test cases as you develop your program (See task 4 below).

Task 2. **Classes solution.**

Add the second part of your project is to handle student results using classes and objects. Each student in the system supposed to take 3 modules in the semester. So, make sure you have created relevant classes including "Student" and "Module". You may can check the sample code given in the Blackboard for your reference. [Sample Code - Task 02 Array of Objects]

Those two classes should capture below list of details

- Student ID *[Length 8] [e.g. w1234567]*
- Student Name
- Module Marks *[1,2,3]*
- Module Grade *[Average >= 80 - Distinction, >=70 - Merit, >= 40 – Pass Else Fail]*

Add an additional option to the menu: "8" to open up few more controls. Otherwise, the program should function as in Task 1

a. Add student name
b. Module marks 1, 2 and 3

Task 3. Add Report.
Add a feature to generate a report out of the system

The menu that is implemented in the task 2 can be updated by adding “c” & “d” to add the following features.

C: Generate a summary of the system which includes

- The total student registrations
- Total no of students who are scored more than 40 marks in Module 1, 2 and 3.

D: Generate complete report with list of students includes

- Student ID
- Student Name
- Module 1 marks
- Module 2 marks
- Module 3 marks
- Total
- Average
- Grade

Sort the list of student details based on average marks highest to lowest.
Note: Extra marks will be awarded for using **Bubble sort**.

Task 4. **Testing**.

Create a table of test cases showing how you tested your program (see below for example). Write a brief (no more than one page) discussion of how you chose your test cases to ensure that your tests cover all aspects of your program. Additionally, write a few paragraphs (up to 1 page) explaining which of the solutions (Array or Class) you think is better and why. Think about which is easier to read/understand, which is easier to modify if necessary, and any other comments you can think of.

Test Case	Expected Result	Actual Result	Pass/Fail
Find available seats before the registration	Press “1” to check the number of available slots	Press “1” to check the number of available slots	Pass
Delete student	Press “3” to delete, checked the latest list by pressing “7”, listed without deleted student.	Press “3” to delete, checked the latest list by pressing “7”, listed with deleted student.	X

Note: Solutions should be java console applications (not windows).

Marking scheme

The coursework will be marked based on the following marking criteria:

Criteria	Max for Subcomponent	
Task 1 Three marks for each option (1,2,3,4,5,6,7,8)	24	**(30)**
Menu works correctly	6	
Task 2 Student class works correctly	14	**(30)**
Module class works correctly	10	
Sub menu (A and B works well)	6	
Task 3 Report – Generate a summary	7	**(20)**
Report – Generate the complete report	10	
Implementation of Bubble sort	3	
Task 4 Test case coverage and reasons	6	**(10)**
Writeup on which version is better and why.	4	
Coding Style (Comments, indentation, style)	7	**(10)**
Complete the self-evaluation form indicating what you have accomplished to ensure appropriate feedback.	3	
Totals		**(100)**

Demo: At the discretion of your tutor, you may be called on to give a demo of your work to demonstrate understanding of your solutions. If you cannot explain your code and are unable to point to a reference within your code of where this code was found (i.e., in a textbook or on the internet) then significant marks will be lost for that marking component.