Manchester Metropolitan University

Department of Computing and Mathematics

ASSIGNMENT COVER SHEET

Unit title:	6G5Z1102: Algorithms and Data Structures
Assignment set by:	Matteo Cavaliere, David McLean
Assignment ID:	1CWK100
Assignment title:	Portfolio Exercises
Assignment weighting:	100% of the total unit assessment
Type: (Group/Individual)	Individual
Hand-in deadline:	As indicated on Moodle
Hand-in format and mechanism:	Exercises with online submission
Support:	The webinars and recorded materials are designed to help you to progress towards your assessed work. The unit teaching team is available for help and guidance. Please contact your tutors to book a one-to-one support session to receive personalised support and feedback on your work (check on Moodle for contact details).

Learning outcomes being assessed:

LO1: Apply problem decomposition to solve programming problems.

LO2: Implementation and appropriate application of static and dynamic data structures.

LO3: Implementation and appropriate application of a variety of programming techniques including pointers, recursion, and algorithms for a variety of problems

LO4: Apply software development techniques including implementation of appropriate software components to enable the modelling of novel problems.

LO5: Analyse and Compare the complexity of algorithms

Extensions

Please note that individual tutors are unable to grant extensions to coursework. Extensions can only be granted on the basis of a PLP, or approved Exceptional Factors (see below).

Exceptional Factors affecting your performance: see Regulations for Undergraduate Programmes of Study (https://www.mmu.ac.uk/academic/casqe/regulations/assessment/docs/ug-regs.pdf). For advice relating to exceptional factors, please see the following website: https://www2.mmu.ac.uk/student-case-management/guidance-for-students/exceptional-factors/ or visit a Student Hub for more information.

Plagiarism

Plagiarism is the unacknowledged representation of another person's work, or use of their ideas, as one's own. Manchester Metropolitan University takes care to detect plagiarism, employs plagiarism detection software, and imposes severe penalties, as outlined in the Student Handbook (http://www.mmu.ac.uk/academic/casqe/regulations/docs/policies_regulations.pdf and Regulations for Undergraduate Programmes (http://www.mmu.ac.uk/academic/casqe/regulations/assessment.php). Bad referencing or submitting the wrong assignment may still be treated as plagiarism. If in doubt, seek advice from your tutor. **As part of a plagiarism check, you may be asked to attend a session with the unit leader, or another member of the unit delivery team, where you will be asked to explain your submitted work**

Assessment Criteria:	Indicated in the attached assignment specification (below).
Formative Feedback:	Formative personalised feedback can be received by booking a one-to-one support session with the tutors (please see Moodle for details)
Summative Feedback Format:	Summative feedback is provided after the hand-in deadline in the form of a written comment

Introduction

This assignment will establish your ability to implement static and dynamic data structures with the opportune classes and methods, writing reusable code using generics where possible. This assignment will also assess your ability to implement advanced data structures, for example trees and graphs, and assess your ability to study the time complexity of algorithms and to implement efficient algorithms for optimisation problems and for sorting. Finally, the assignment will assess your ability to use specific features of languages (such as pointers in C++) to design efficient algorithms.

Specification of 1CWK100

The assignment, 1CWK100, will assess your ability to use classes, instances and inheritance in C#, with the IDE Visual Studio, to implement stacks and queues. Moreover, the assignment will evaluate your ability to use dynamical data structures, such as linked lists, and recursion, a fundamental technique, to successfully implement binary trees and binary search trees, two examples of very important data structures. This assignment will also assess your ability to implement more advanced data structures, such as AVL trees, which allow for very efficient implementations of searching algorithms. The assignment will also assess your ability to use efficient sorting algorithms, employing graphs (which are advanced data structures used to store data that are characterized by completely arbitrary relations). Finally, the assignment will assess your ability to develop efficient algorithms and use specific features of languages (such as pointers in C++).

Each week you should ensure that you have read and understood all the material, attending the scheduled sessions and completing the corresponding exercises.

The assignment 1CWK100 is comprised of 5 assessed exercises, which correspond to five general topics, studied in the unit. Each one of these 5 exercises includes 2 assessed tasks. The tasks must be solved using the techniques seen in the unit.

The list of exercises is reported below. The full specification of the exercises is on Moodle at the corresponding week where you can find the corresponding file. Each file contains the indication of the two assessed tasks.

Week 1

Introduction to C# and basic static data structures

Week 2

Introduction to dynamics data structures

The assessed exercise is **ExerciseWeek1_2.** This will cover Basic Static & Dynamic Data Structures

Week 3

Binary Search Trees and AVL trees

The assessed exercise is **ExerciseWeek3.** This will cover Trees.

Week 4

Sorting, Complexity and Efficient Greedy Algorithms

The assessed exercise is **ExerciseWeek4.** This will cover Efficient Algorithms for Sorting & Optimization.

Week 5

Graphs

The assessed exercise is **ExerciseWeek5.** This will cover Algorithms on Graphs.

Week 6

Software development, C++, pointers and unit testing

The assessed exercise is **ExerciseWeek6.** This will cover C++ and pointers.

Hand-In Method

The **deadline** for the online submission is on Moodle.

On Moodle **you must submit a single zip file** (up to 100Mb) so to submit the assignment please carefully follow the instructions below:

For each one of the 5 assessed Exercises you should **create a separate folder** (named "ExerciseWeek1_2", "ExerciseWeek3", "ExerciseWeek4", "ExerciseWeek5", "ExerciseWeek6")

Each one of these 5 exercise folders should contain 2 subfolders **which correspond to the two assessed Tasks** of the exercise (eg., you can name these two subfolders as "Task A" and "Task B").

Each one of these subfolders (e.g., "TaskA" and "TaskB") should contain:

- Your **solution directory** including the **executable and all solution files** of the Task

- A "Support Document" (can be a Word file) which contains the **Onedrive link** to the **screencast** (see below "How to create a Onedrive link to your screencast"). Do not try to add the screencast file directly in the folder as may be too large for the max size allowed by Moodle and you will not be able to submit it.

- A "Critical Reflection" document (can be a Word file).

The 5 Exercises folders ("ExerciseWeek1_2", "ExerciseWeek3", "ExerciseWeek4", "ExerciseWeek5" and "ExerciseWeek6") must be added into a **single zip file** which needs to submitted at the submission point on Moodle (1CWK100 – 1 Portfolio 100%).

Feedback & Grades

We will be working towards the elements of this assessment during each session and we strongly encourage you to work on the exercises as soon as they become available on Moodle.

Formative feedback is provided by the tutors. You can improve your work (**before you hand it in online on Moodle**) based on the formative feedback received. Please contact your tutors to book a one-to-one support session to receive personalized formative feedback.

Grades and summative feedback will be released after the hand-in deadline. You will receive summative feedback in the form of a written comment highlighting your performance against the marking criteria.

Marking Criteria

Each one of the 5 assessed exercises will receive a grade 0 - 100. The overall grade for the assignment 1CWK100 is the average grade obtained by considering the grades of the 5 assessed exercises (the 5 assessed exercises have equal weight).

Each one of the 5 assessed exercise contains 2 assessed tasks.

Please notice that a submitted task is considered for grading only if the corresponding screencast provides a descriptive and adequate description of the submitted task. See the info below: "What should the Screencast contain?".

The grade of each assessed exercise depends on the tasks completed and the quality of the screencast and of the critical reflection. The addition of the critical reflection is never detrimental but can possibly increase your grade (see below).

The grade for each assessed Exercise is determined in 2 steps.

Step 1.

The grade is determined depending on the number of tasks partially / mostly / fully completed.

0 – 20 One of the two tasks has been partially completed (only a part of the task has been correctly attempted). The other task has not been attempted.

20 – 39 Both tasks have been partially completed.

40 – 50 One of the two tasks has been mostly completed (the entire task has been correctly attempted but there are some minor flaws). The other task has not been attempted or only partially completed.

50 – 60 One of the two tasks has been fully completed (the entire task has been correctly completed). The other task has not been attempted or only partially completed.

60 – 70 Both tasks have been mostly completed.

70 – 80 One of the two tasks has been fully completed. The other task has been mostly completed.

80+ Both tasks have been fully completed. Evidence of algorithmic efficiency, code reusability and maintainability, code readability and low cyclomatic complexity, usability and validation.

Step 2

If you have added to your submission the **document of critical reflection** then this is used to possibly increase the grade obtained at step 1 (but still **keeping it within the grade band determined at step 1**) depending on the quality of your critical reflection (see the info below: "What the Critical Reflection document should contain?"). Please note that the inclusion of the document of critical reflection **is never detrimental** for your grade but will only be used to possibly increase the grade obtained at step 1.

The Screencasts

What should the Screencast contain?

It is expected that the screencast would contain:

- Explanation of the code organization for the task: methods used, class members and class hierarchy - when classes have been used (If a part of the task does not involve coding, simply explain how you have solved that part of the task).
- Explanation of the key parts of the code and of the programming techniques used to solve the task
- Explanation of the logic behind your proposed solution and links of your implementation with the theory /methodology seen in the unit (how the theory / methodology seen in the unit has been used in your implementation)
- Discussion of how confident you are that your solution works and successfully solves the task (e.g., could be a presentation of the code execution).

You can be creative in your screencast and make reference to any material you think may help your explanation of the submission (e.g. slides of the lectures).

The screencast should have either:

screen recording with audio OR **screen recording without audio but with appropriate subtitles or annotations.**

How long should the screencast be?

As a general guideline it is expected that a screencast should be max 8 minutes for each submitted task (this means max 16 minutes if you have submitted both assessed tasks present in an exercise). However, this is just a guideline and there is no penalty for overlong screencasts but consider you may penalise yourself if you are not succinct.

How to record a screencast?

You can use any software which allows to record the screen and the audio. For example, you could use OBS, available at the following link: https://obsproject.com/download

If you cannot find any available software, you can also record the screencast with your phone.

How do I create a OneDrive link to my screencast?

Once you have created your screencast, you should upload it on OneDrive, get the corresponding link which you will need to add to the Support Document in your submission.

You can read more Info on OneDrive here

https://www.mmu.ac.uk/isds/support/apps/office/onedrive/

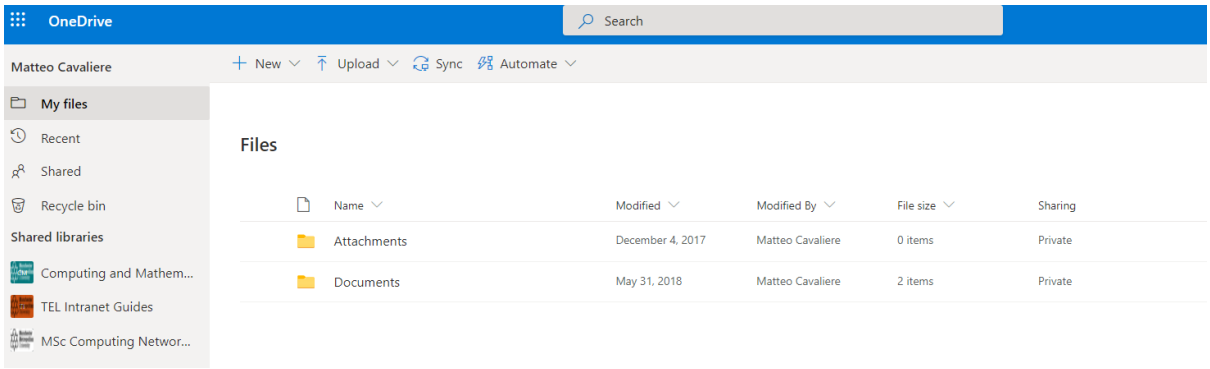
One possible way **to get the link** to your screencast is described below

Accessing your OneDrive account. You can access your OneDrive folder via Office365

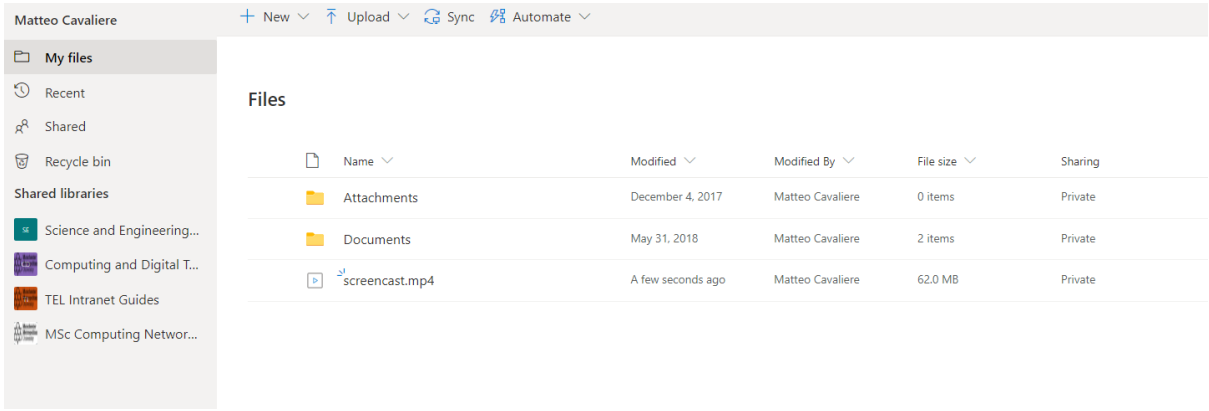
at http://office365.mmu.ac.uk/

and you will need to input your Manchester Met credentials (username is your student number, the password is your regular network password you use to log in on your machine)

This will get you here :



Use **Upload** to upload your screencast video.



Then right-click on the screencast file. Click on Share. In “People you specify can view” choose “Anyone with the link” and click on “Apply”. Then “Copy Link”. **Copy the obtained link** in the Support Document that you add in your submission.

The Critical Reflections

What should the critical reflection document contain?

There are a few formal models of critical reflection (such as the "Gibbs Reflective Model"*), which you may choose to use but for this assignment, though it is not necessary to use a formal reflection model as long as your critical reflection is well organised.

In your critical reflection you should focus on the difficulties you have found during the solution of the task and identify any factors that have helped you to tackle them. Also try to think critically of your submission: If you are not convinced about some aspects of your submission explain why you think this is the case and how you think such aspects may be addressed. Feel free to add images (e.g. screenshots) in your document.

A possible structure for your critical reflection:

- Describe and analyse the positive/negative aspects of your submission (e.g., describe things which went well or were particularly easy and things which you found more difficult during the solution of the task)
- Focus on some of the key difficulties you have found and describe what helped you to overcome the difficulties (could be specific aspects of the theory, feedback received, etc..). You can also talk about your general understanding of the theory / methodology behind the solution of the specific task.
- Is there anything which you think may be improved in your submission? (e.g., complexity of your code, efficiency, … or any other aspect of your submission which you think may be improved). Discuss how this could be improved and any skill(s) you may need to develop in future to improve those aspects.

*G. Gibbs. Learning by Doing: A Guide to Teaching and Learning Methods. 1988.

A general guideline for the evaluation of the critical reflection:

Excellent

A coherent, creative analysis of the positive and negative aspects of your development for the submission. A complete presentation of the key difficulties found and a careful description of what helped you to overcome those difficulties. A complete identification of the aspects of your submission which could be improved and a detailed description of possible approach to improve those aspects. The level of detail provided is relevant and, at the same, succinct.

Satisfactory

An adequate analysis of the positive and negative aspects of your development for the submission. A sufficient presentation of the key difficulties found and there is some description of what helped you to overcome those difficulties. An adequate description of the aspects of your submission which could be improved and some details on the possible approach to improve those aspects. The level of detail provided is sufficient.

Unsatisfactory

An incomplete analysis of the positive and negative aspects of your development for the submission. Little or no presentation of the key difficulties found and there is little or no description of what helped you to overcome those difficulties. A very limited description of the aspects of your submission which could be improved and little or no details on the possible approach to improve those aspects. The level of detail provided is insufficient.

How long should the critical reflection be?

It is expected that your critical reflection should be a maximum of 500 words for each task submitted (this means max 1000 words if you have submitted both assessed tasks present in an exercise). However this is just a guideline and there are no penalties for overlong critical reflections, but consider whether you may penalise yourself if you are not succinct.