INTI International College Penang **School of Engineering and Technology**

3+0 Bachelor of Science (Hons) in Computer Science, in collaboration with Coventry University, UK

3+0 Bachelor of Science (Hons) in Computing, in collaboration with Coventry University, UK

Coursework cover sheet

Section A - To be completed by the student

Full Name:	
CU Student ID Number:	
Semester:	
Lecturer: Teng Wei Jian	
Module Code and Title: 5003CEM Advanced Algorithms	
Assignment No. / Title: Individual Report	67% of Module Mark
Hand out date: 17 October 2023	Due date: 19 November 2023
Penalties: No late work will be accepted. If you are unable to submit coursework on time due to extenuating circumstances you may be eligible for an extension. Please consult the lecturer.	
Declaration: I/we the undersigned confirm that I/we have read and agree to abide by the University regulations on plagiarism and cheating and Faculty coursework policies and procedures. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for plagiarism checking. Signature(s): -------------------------------------	

Section B - To be completed by the module leader

Intended learning outcomes assessed by this work: LO1: Understand and select appropriate algorithms for solving a range of problems and reason about their complexity and efficiency. LO2: Design and implement algorithms and data structures for novel problems. LO3: Understand the intractability of certain problems and implement approaches to estimate the solution to intractable problems. LO4: Describe the issue of data consistency in non-synchronous applications. LO5. Design and implement a basic concurrent application.		
Marking scheme	Max	Mark
Refer to attached rubric		
Total		
Lecturer's Feedback		

Internal Moderator’s Feedback

Instructions

1. There will be FOUR (4) programs that must be completed and submitted before the stated due date. The programs will cover a range of data structure and algorithms taught in this module as well as a basic concurrent application.

   You are required to submit the commented source code (in either Pycharm or dev C++ project) of your solutions in zip format.

2. Prepare a documentation (in pdf or word format) for the programming tasks and submit before the due date. The documentation should include:
   - Screen capture for each program output along with explanation of the code. If the questions asks for discussion, you can include them here as well.
   - The weakness of your solution and/or what is not working in your solutions.
   - Reflection – challenges and what you have learnt from this coursework.
   - References – list of all the references that you referenced in this assignment including code references as well.
   - Appendix – List of your solutions (code). No screen capture is allowed for this section. You must copy all your code in text form and include them here.

3. There will be a private VIVA session for each student to allow you to demonstrate your understanding and highlight the originality of your work. You are required to explain how your code works. The instructor will also ask you questions about your code.

4. Read the important notes and marking rubric at the end of this document carefully to avoid any penalties.

Programming Tasks

Question 1: Pizza Ordering System

Design and develop a Pizza Ordering System based on the following requirements:

- The program should be menu driven, giving the user various choices of operation that allows the user to:
    - place an order for the pizza(s).
    - View their order details.
    - Modify or delete a particular order if necessary.

- For every order, the following information will be stored:
    - OrderID (this should be autogenerated), Pizza Code, toppings, size, unit price, quantity and customer information.
    - The customer information should consists of customerID, name, address and contact number.
    - The system shall display additional information such as amount (unit price * quantity) when viewing the order details.

- The program must use BST data structure to facilitate each operation.
- The system shall demonstrate a good OOP design, data validation and error handling.

Question 2: Hash Tables

Devise an experiment that will determine which of the two hash function defined below is “better”.

You first need to define your 2 hash functions. Start by choosing 2 random numbers a and b such that $a < b$. We will also assume that the table will have a size of 5500. For a key of i, the 2 functions will be:

- $h(i) = (ai + b) \ mod \ 5500$
- $h(i) = floor\left(5500 * \left(\left(i * \frac{a}{b}\right) mod \ 1\right)\right)$

Write a program that will insert 5000 unique and randomly generated numbers into two separate hash tables using the functions define above. In the case of a collision use open addressing linear probing to handle it. Use suitable methods to investigate which function performs better. Your experiment should be repeated at least 10 times.

Using the observations and results obtained, write an argument to support your findings. You may use other additional methods of investigation as well to make a more compelling argument.

Question 3: Graph

Write a program that uses graph concepts. The program should fulfil the following requirements:

- Construct a weighted directed graph class that consists of the following methods:
    - listAdjacentVertex: for a given vertex, this method should list all adjacent vertices.
    - heaviestVertex: for a given vertex, this method should which of the adjacent vertices have the highest weight.
- Create two different graph objects (with at least 8 vertices, 30 edges and random weights) and call the above two functions and display the results.

Explain the method/design you used to represent the graph structure in your program.

Question 4: Concurrent process

a) Write a function to calculate and display the car loan monthly repayment. Assume a flat interest rate is used.

   You may refer to this link to understand how the calculation of monthly repayment works: https://www.comparehero.my/personal-loan/articles/heres-how-car-loans-work-and-why-interest-charges-are-higher-than-you-think

b) Write a program to allow users to calculate the monthly repayment for 3 different customers. The program should call the function defined in (a) concurrently.

Observe the output of your solution and discuss the relevant theories that are used.

Important Notes

- Label your question number clearly in your documentation
- In the appendix section, no print screen of codes is allowed. No marks will be awarded to you if your codes are in image form and cannot be checked for plagiarism. Any suspected plagiarism will be report and will follow the academic dishonesty procedures.
- You will get a penalty if your similarity is more than 30%
- Submit you work in canvas before the deadline. Late submissions will be given zero marks.
- Viva is required to assess your understanding of your solutions and the originality of your work. If you fail to attend the viva, you will be given zero marks for all the components labelled under viva in the rubric below (total of 50/100 marks).

Marking Rubric:

	CODE SUBMISSION 40 MARKS	**Documentation 10 marks**	**VIVA 50 MARKS**		
Marking criterion	Code 40	Documentation 10	Knowledge of code 30	Critical reflection 10	Clarity of communication 10
1st (70+)	Code is fully working, bug-free, well-commented, possibly with some advanced features.	Each section in the documentation are organised with relevant title/sub title. Discussions are very comphrehesive and in depth.	Student shows sophisticated and detailed knowledge of how the code works at every level. Student can explain the design and implementation decisions.	Able to critique the implementation and offer a range of alternative possibilities; will be able to explain implementation decisions at a sophisticated level, including complexity analysis.	Able to speak clearly and coherently in a well-prepared way, and without the need for frequent prompting. Authoritative on the material. Great question-handling.
2.1 (60-69)	Code works, may have one or two non-important bugs. Commented. Does not has any advanced features.	Each section in the docuemntation are mostly organised with mostly relevant title/sub title. Discussions are somehow comphrensive with very little areas that lack of details explanation	Good knowledge of how major sections of the code work. Is able to explain design decisions and why things were implemented in particular ways.	Able to critique the implementation against possible alternatives (which may be limited to one). Can broadly explain implementation decisions, but less effective on detail. May be less effective on complexity analysis.	Generally able to speak clearly and coherently without the need for many prompts. Clear ability to communicate what's been done. Most questions well handled.

2.2 (50-59)	Code generally works, may have bugs, and may not cover all the required features even at basic level. Has some comments.	Each section in the docuemntation are organised with mostely relevant title/sub title. Some discussions are moderate with some discussion are in depth and some lack of details.	Some knowledge of how the code works but there may be areas of confusion. Ability to explain basic decisions about the design and implementation but again there may be misconceptions.	Less able to be critical about the code, and may not be able to discuss alternative implementations except in broad terms. May show confusion on complexity analysis.	Able to say something and can answer questions but may be hesitant. May need some prompting, but insight shown. Some questions may not be well answered and some answers can be wrong.
3rd (40-49)	Code may not work; may not cover basic required features, may have significant bugs; but shows potential to work. May have some comments.	Each section in the docuemntation are mostly organised with some relevant title/sub title, with some topic not included. A few discussions are comprehesive, but some areas lack of details.	Limited knowledge of how the code works which is unlikely to show insight at detailed levels. Aware of the design of the code but not much insight into different design decisions that could be made.	Unlikely to be critical about the code beyond describing what it does. Little awareness of possible alternatives or complexity.	Halting, hesitant. May show some elementary insight but unable to speak authoritatively. May be able to deal with a few questions but struggle with most.
Fail (0-39)	Code is incomplete, does not work, is not commented, does not show clear potential. At the zero end, nothing has been done.	Documentation is very brief with many section incomplete/no deliver. At the zero end, nothing has been done.	Little if any ability to explain how the code works. May be attempting to explain by reading through code and trying to work it out during the viva, where the results are largely or totally wrong / incomplete. At the zero level the explanation (if any) will have no value.	Largely unable to be critical about the code or implementation. At the zero end, student will be completely unable to evaluate the code.	Little knowledge or engagement with the module and the performance shows this. May be attempting to 'make up' answers on the spot. At the zero end, student may appear to be unaware of the problems set.