



DESCRIPTION OF COURSEWORK

Course Code	CST207
Course Name	Design and Analysis of Algorithms
Lecturer	Dr. Mohammed N. M. Ali
Academic Session	2024/04
Assessment Title	Project (Group)

A. Introduction/ Situation/ Background Information

This project evaluates the student's ability to select a searching and sorting algorithm to solve a problem and explain their selection by analyzing and comparing different algorithms.

B. Course Learning Outcomes (CLO) covered

At the end of this assessment, students are able to:

CLO 3 Demonstrate problem-solving skills via concepts, techniques, and proficiencies in algorithm analysis which leads to potential applications in the self-development stage e.g., further studies.

C. University Policy on Academic Misconduct

1. Academic misconduct is a serious offense in Xiamen University Malaysia. It can be defined as any of the following:

- i. **Plagiarism** is submitting or presenting someone else's work, words, ideas, data or information as your own intentionally or unintentionally. This includes incorporating published and unpublished material, whether in manuscript, printed or electronic form into your work without acknowledging the source (the person and the work).
- ii. **Collusion** is two or more people collaborating on a piece of work (in part or whole) which is intended to be wholly individual and passed it off as own individual work.
- iii. **Cheating** is an act of dishonesty or fraud in order to gain an unfair advantage in an assessment. This includes using or attempting to use, or assisting another to use materials that are prohibited or inappropriate, commissioning work from a third party, falsifying data,

or breaching any examination rules.

2. All the assessment submitted must be the outcome of the student. Any form of academic misconduct is a serious offense which will be penalised by being given a zero mark for the entire assessment in question or part of the assessment in question. If there is more than one guilty party as in the case of collusion, both you and your collusion partner(s) will be subjected to the same penalty.

D. Instruction to Students

This project is **Group work**. Each group should be constructed with **2-5** members (**Maximum 5 members**). You also have the choice to do it individually. Each group should submit a project report using a file (C.c or C++.cpp) and a pdf report:

- (i) “<Student_Name>_<Student_ID>_CST207_Project_Group_202404.pdf”
- (ii) “<Student_Name>_<Student_ID>_CST207_Project_Group_202404(.c or .cpp)”

This report should contain all the source code, running results, and analysis and justification of the chosen algorithms.

Please attach the **Cover page** and **Marking Rubrics** with your Project in **.pdf** file. The deadline is **18:00, 8th July 2024**. An overdue penalty will be applicable to the project that is submitted after the deadline.

* Your codes will be sent to a **Plagiarism** detection system for duplication checking. Please write your codes independently. (Modify your code if you copy some fragment from the Internet because your classmates may copy the same fragment.)

E. Evaluation Breakdown

No.	Component Title	Percentage (%)
1.	Project	100
	TOTAL	100

F. Task(s)

You are a software developer at a software development company. One of the largest car dealerships has approached your company, requesting a program to manage their business operations. The purpose of this system is to streamline tasks related to car brands, customers, billing, and more.

The system should include the following functionalities:

- Add a new car
- Modify car data
- Display all cars
- Sort cars by brand or price
- Search for a car by its unique ID
- Track the number of cars sold
- Remove a car record
- Sort cars by best-selling brand

You are required to develop an algorithm using either C or C++ to implement this system, ensuring it meets the following specified requirements:

1. Implement a secure login screen that permits access only to authorized personnel.
2. Upon successful login, the system should present the user with a list of options for various tasks they can perform.
3. When adding a new car, the system should generate a **random car ID**. This ID should start with the first three letters of the car brand. **For example**, a Proton car might have an ID like PRO202103929.

Use a random function to create these IDs.

Menu:

4. The system must be able to save car information, including the randomly generated ID, brand, color, country of manufacture, year of manufacture, and price.
5. Include a search function that allows users to look up cars by their ID. This should be one of the available tasks. You can implement the search using either the **Linear Search** or **Binary Search** algorithms.
6. Provide an option to display an unsorted list of all car records in the company's database.
7. Allow users to view car records sorted by ID, price, or brand. Implement sorting using either **Merge Sort** or **Quicksort**.
8. Enable users to search for the best-selling cars, sorted by brand name. Use **Radix Sort**, **Bubble Sort**, or **Selection Sort** for sorting, and **Binary Search** for searching.
9. When a car is purchased, the system should generate a bill for the customer. This bill must include the customer's name, customer ID, car ID, car brand, year of production, color, price, any applicable discount, and a unique random purchase order number. It should also display the **total price** of the purchase.
10. Users should be able to search for issued bills by customer ID or bill date. Implement this search using the **Linear Search** algorithm.
11. The system should generate a report of all company purchases within a specified period. This report should include car details (ID, brand, color, customer name, exact date, and price) and the total price of all sold cars. Use a **binary search algorithm** to compile this list, and sort the items by price using **Quicksort**.

After building and running your algorithm, you are required to complete a report explaining in brief the following:

1. The search and sort algorithms implemented in the system.
2. In your opinion, justify which is the best search algorithm you have applied.

3. In your opinion, justify which is the best sorting algorithm you have applied.
4. Briefly explain the time and space complexity of each algorithm used in the system.
5. Search the internet for two additional search algorithms. Provide a brief description of each and compare them to linear and binary search algorithms.
6. Search the internet for two additional sorting algorithms. Provide a brief description of each and compare them to merge sort and quicksort sort algorithms.
7. Expand on your answers using online resources. Where possible, support your answers with citations and list the references at the end of your report.
8. Include the code of your system in the appendix section of your report.
9. To assess the personal skills of each student in the group, specify the contribution of each student in your report.
10. Note: You are free to design the system as you see fit; however, you must apply the required algorithms as specified above.

APPENDIX 1

MARKING RUBRICS

Component Title						Percentage (%)		
Criteria	Score and Descriptors					Weight (%)	Marks	
	Excellent (90-100)	Good (80-89)	Average (60-79)	Need Improvement (50-59)	Poor (0-49)			
Quality of implementation	Program runs efficiently without errors. Adheres to good programming practices.	Provide more than 50% correct code. Somewhat adheres to good programming practices.	Provide up to 50% correct code. Room to improve on good programming practices.	Attempted to complete code but with many logic errors. Some application of good programming practices.	Unable to provide any correct code. Poor/no application of good programming practices.	30		
The implementation and the justification of choosing the search algorithm	Implement the algorithm correctly with convenient justification. Solutions are properly organized, with accurate and great details.	Implement the algorithm correctly with convenient justification. Solutions are properly organized and with sufficient details.	Implement the algorithm and justify. Some are correct	Inaccurate implementation and justification.	Use different algorithms, very little work, or no justification.	20		
The implementation and the justification of choosing the sorting algorithm	Implement the algorithm correctly with convenient justification. Solutions are properly organized, with accurate and great details.	Implement the algorithm correctly with convenient justification. Solutions are properly organized and with sufficient details.	Implement the algorithm and justify. Some are correct	Inaccurate implementation and justification.	Use different algorithms, very little work, or no justification.	20		
Report Format	The report is well written without grammar and spelling mistakes. Good presentation format. All sections are present. A good set of references.	Report format adhered to. Grammar is mostly sound. All sections are present. Some references.	Some grammatical mistakes. The formatting needs improvement. Minimal references.	Poor format. Plenty of grammatical mistakes. Minimal or no references.	The format was completely ignored. Major inconsistencies and/or no references.	15		
The assessment of each student's contribution to the project.	The highest contribution student	The next highest contribution student	The medium contribution	The lowest contribution	The student did not have any contribution	15		
TOTAL						100		

Note to students: Please print out and attach this appendix together with the submission of coursework