

Course Learning Outcomes:

Upon completion of this assignment, you should be able to:

CLO1	Explain the essential facts, concepts, principles, strategies, and theories relating to Information Technology applications. (C2, PLO1)	Class Test
CLO2	Demonstrate intellectual independence, logical and analytical thinking skills to develop creative and innovative solutions for a range of Information management and IT problems. (C3, PLO2)	Individual Assignment
CLO3	Communicate effectively and professionally with peers, clients, superiors and society at large both in written and spoken form. (A3, PLO5)	Individual Assignment

1.0 INDIVIDUAL ASSIGNMENT DESCRIPTION

DOSTA COFEE SHOP ORDERING SYSTEM

Dosta is a new coffee shop launched in Kuala Lumpur, Malaysia recently after the covid outbreak. Dosta branches sell hot and cold drinks, sandwiches, cakes and pastries, and snacks. As its business is expanding they approach your Software Development company to develop an Ordering System to handle all the primary information required to maintain customer orders. Once the system is launched, staff will have to login to enter the main screen and customers must register first to place their order.

The system will ask the users to choose whether they are staff, first time customers, or registered customers.

- a. **Staff**
 - add new menu item with related details
 - update the menu item
 - create the customer account
 - search for any customer order based on customer ids
- b. **First time customers should register their account (mandatory)**
- c. **Registered customers**
 - modify the customer details
 - place order and make the payment
 - cancel any order placed
 - receipts will be generated at the end of the ordering process

The system submitted should compile and be executed without errors. Besides, validation should be done wherever needed. Customer, item and order details need to be stored in text files. Primary Key should be implemented, no duplication of same customer, item or orders should be accepted.

2.0 REQUIREMENTS

- i. You are required to carry out extra research for your system and document any logical assumptions you made after the research.
- ii. Your program should use symbolic constants where appropriate. Validations need to be included to ensure the accuracy of the system. State any assumptions that you make under each function.
- iii. You are required to store all data required for your system in separate text files.
- iv. You are expected to use control structures, functions, array, pointers, structures, unions, and files in your program. Your program must embrace modular programming technique and should be menu driven. Functions of similar operations can be grouped (or kept alone) and stored as separate C files. Header files are to be stored separately as .h files.
- v. You may include any extra features which you may feel relevant and that add value to the system.
- vi. There should be no need for graphics (user interface) in your program, as what is being assessed, is your programming skill not the interface design.
- vii. You should include the good programming practice such as comments, variable naming conventions and indentation.
- viii. In a situation where a student:
 - ***Found to be involved in plagiarism, the offence will be dealt in accordance to APU regulations on plagiarism.***
- ix. You are required to use portable ANSI C programming language to implement the solution. Use of any other language like C++/Java etc. is not allowed.
- x. Results of a comprehensive testing is to be included in your document in the form of Input/Output screenshots with sufficient explanation. The tests conducted shall take into consideration of all valid inputs and negative test cases.

3.0 DELIVERABLES

You are required to submit:

- i. A softcopy of the program coded in C – submitted in Moodle. The program should include the following:
 - Basic C concepts such as displaying and reading of text, variables, and assignment of values, comments – to explain various parts of the program, etc.
 - Intermediate C concepts such as control structures – selection and iteration control structures, use of arrays – single / double scripted, string.
 - Advanced C concepts such as functions – programmer defined and library functions, pointers, structures, unions, linked list, and files.
 - Any other features of C that has not been covered.
- ii. A documentation of the system, that incorporates basic documentation standards such as header and footer, page numbering and which includes
 - Cover page
 - Table of contents
 - Introduction and assumptions
 - Design of the program – using pseudocode **and** flowchart – which adheres to the requirements provided above
 - Additional features which have been incorporated in the solution in terms of design and C codes (sample segment of source code from the system created)
 - Sample outputs when the program is executed with some explanation of the outputs / sections of the program
 - Conclusion
 - References – APA Referencing
- iii. Files to be uploaded to Moodle (ONLY FOLLOWING 3 FILES):
 1. **Documentation file** (.pdf)
 2. **Program / Source files** (.c files), **Header files** (.h files), **and text files**. (all compressed as single .zip or .rar file)
- iv. Submission
 - All three files to be uploaded to Moodle on or before November DD/MM/22:11:59 PM.

4.0 ASSESSMENT CRITERIA

- | | | |
|------|---|------------|
| i. | <u>Design solution (Pseudocode and Flowchart)</u> | <u>20%</u> |
| | Detailed, logical and application of appropriate idea. | |
| ii. | <u>Coding / Implementation</u> | <u>30%</u> |
| | Appropriate application of C concepts (from basic to advance), good solution implemented with validation and met all the requirements with additional features. | |
| iii. | <u>Documentation</u> | <u>20%</u> |
| | Overall standard and layout, referencing (Harvard), Input/Output screen capture and assumptions. | |
| iv. | <u>Demonstration</u> | <u>20%</u> |
| | Know how to execute and able to trace the system. | |
| v. | <u>Question and Answer</u> | <u>10%</u> |
| | Answered the questions based on the assignment submitted during presentation. | |

5.0 PERFORMANCE CRITERIA

Distinction (75% and above)

This grade will be assigned to work which meets all requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of C concepts up to advanced level. The program solution is unique with excellent coding styles and validation. The program implemented maps completely against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with hardly any errors / omissions. The documentation does not have any missing components. Sample outputs documented have clear explanation. All work is referenced according to Harvard Name Referencing convention. Student must be able to provide excellent explanation of the codes and work done, show additional concepts / new ideas used in the solution, able to answer all questions posed with accurate / logical answers / explanation provided with sound arguments and clear discussion. Overall an excellent piece of work submitted.

Credit (65%-74%)

This grade will be assigned to work which of good standard and meets most of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of C concepts up to at least intermediate level. The program solution is unique with good coding styles and validation. The program implemented maps well against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with minor errors / omissions. The documentation does not have any missing components. Sample outputs documented with some explanation. All work is referenced according to Harvard Name Referencing convention but with some minor errors / omissions. Student must be able to provide good explanation of the codes and work done, answer most questions posed with mostly accurate / logical answers / explanation. Overall a good assignment submitted.

Pass (50%-64%)

This grade will be assigned to work which meets at least half of the basic requirements (approximately 50%) stated in the questions. The program runs smoothly when executed. There is clear evidence and application of C concepts at basic level. The program solution is common with basic coding styles and validation. The program implemented somewhat maps with the design (pseudocode and flowchart) as seen in the documentation. The design of the solution is average in terms of logic and style with some errors / omissions. The documentation has some missing components. Sample outputs documented but without any explanation. Did some referencing but not according to Harvard Name Referencing convention and with some minor errors / omissions. Student must be able to explain some codes and work done and able to answer some questions posed with some accurate / logical answers / explanation. Overall an average piece of work submitted.

Fail (Below 50%)

This grade will be assigned to work which achieved less than half of the requirements stated in the question. The program is able to compile but not able to execute or with major error. The program solution has only basic coding styles with no validation. The program solution has little or no mapping with the design. The design of the solution has major / obvious errors / omissions. The documentation has some missing essential components. No referencing. Student is barely able to explain the codes / work done and answer given on the questions posed but with mostly inaccurate / illogical answers / explanation. Overall a poor piece of work submitted.