sCourse 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)**	**Presentation**

1.0 INDIVIDUAL ASSIGNMENT DESCRIPTION

APU Programming Café Management System

In order to improve the students' performance, enhance their problem-solving skill and confidence level in coding, APU decided to set up an experimental program called "Programming Café" to provide additional coding sessions for students outside their regular timetable. Every session will have a duration of TWO hours and classroom will be assigned for the purpose.

Programming Café sessions will be conducted by selective APU Graduate Students with good knowledge in their respective programming language, who will be paid a flat rate of RM100 for each TWO hours Programming Cafe session conducted.

APU has selected some of the common programming languages for the Programming Café Sessions in this experimental phase. The selected programming language sessions are listed in Table 1. Additional session on different coding subject can be added as required. However, all the sessions listed in Table 1 must be added into the system by default.

Table 1: Programming Café Sessions

Session Code	Title	Day	Start Time	Location	Tutor Code
PYP101	Python Programming	Saturday	9.00am	C-01-01	T01
JAV102	Java Programming	Sunday	9.00am	C-01-02	T02
CPL103	C Programming	Saturday	2.00pm	C-01-03	T03
WEB104	Web Development	Sunday	2.00pm	C-01-04	T04
CSP105	C Sharp Programming	Monday	7.00pm	C-01-05	T05

Table 2. Tutor

Tutor ID	Name	Title
T01	Albert	Web Development
T02	Amad	C Sharp Programming
T03	Steve	Python Programming

You are required to write a program in C language for APU to manage the Programming Café Sessions based on the requirement given below. The system must be able to handle THREE different types of users in their respective functional areas. The three types of users are Admin, Tutor, and Student. Relevant features associated with the user type will be provided in your system automatically based on the login credential.

1. Functional features for Admin user type

a) Registration of Tutor

Admin must be able to add new tutor into the system. Details listed in Table 2 need to be recorded in the process. Any additional details can be added if necessary. All the tutors listed in Table 2 to be added by default and available for selection when the system is run for the first time. The registration process will also include tutor password allocation to allow the tutor to login using **Tutor Code** and **Password** allocated.

b) Adding new programming café session.

System user must be able to add new session or title into the system. All the sessions listed in Table 1 to be added by default and available for selection when the system is run for the first time. The tutor must be registered first before a new session can be assigned to the

tutor. Each session can only be assigned to one tutor and one tutor can only take up one session.

c) Registration of Student

Basic student details such as TP number and name must be recorded in the registration. The registration process will also include student password allocation to allow the student to login using **TP Number** and **Password** allocated.

d) Enrol student in a session.

A session can be linked to the student in this process.

e) Listing of Programming Café sessions and participating students. See below for the sample output for this option:

Student Name	Session Code	Tutor Code	Location
Abdullah	CSP105	Amad	C-01-05
Yogeswaran	PYP101	Albert	C-01-01

2. Functional features for Tutor user type

Allow Tutor to view listing of Session(s) assigned to their Tutor Code. Listing format refer to 1 (e).

3. Functional features for Student Allow student to view listing of Session(s) assigned to their student's name. Listing format refer to 1 (e). In addition, students are allowed to enrol themselves in any of the available session.

4.0 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 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.

iv. You may include any extra features which you may feel relevant and that add value to the system.

v. 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.

vi. You should include the good programming practice such as comments, variable naming conventions and indentation**.**

vii. In a situation where a student:
- ***Failed to attempt the assignment demonstration, overall marks awarded for the assignment will be adjusted to 50% of the overall existing marks.***
- ***Found to be involved in plagiarism, the offence will be dealt in accordance to APU regulations on plagiarism.***

viii. You are required to use portable ANSI C programming language to implement the solution. Use of any other language like C++/Java and etc. is not allowed. Global variable is not allowed.

ix. 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. Sufficient test data is required in text files.

2.0 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 **OR** 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 using 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 by **DD/MMM/YYYY** latest by **11.30PM**.

3.0ASSESSMENT CRITERIA

i. Design solution (Pseudocode and Flowchart) 20%

Detailed, logical and application of appropriate idea.

ii. Coding / Implementation 30%

Appropriate application of C concepts (from basic to advance), good solution implemented with validation and met all the requirements with additional features.

iii. Documentation 20%

Overall standard and layout, referencing (Harvard), Input/Output screen capture and assumptions.

iv. Demonstration 20%

Know how to execute and able to trace the system.

v. Question and Answer 10%

Answered the questions based on the assignment submitted during presentation.

4.0PERFORMANCE 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.