



DESCRIPTION OF COURSEWORK

Course Code	CST103
Course Name	Programming in language C
Lecturer	Prof. Li Xiaochao and Prof. Yang Chenhui
Academic Session	2023/09
Assessment Title	Lab Report and Presentation

A. Introduction/ Situation/ Background Information

This task evaluates the student overall abilities around C programming. Student is asked to develop a software for himself/herself, that is, named as “knowledge point learning tracking system”, or KPLTS for short, a software that can help learn CST103. The student is the target user of this software, so he/she can go through the user requirement analysis, conceptual design, design, coding, debugging, optimization to documentation, demonstration and presentation. It will cover all the CLOs of CST103 and a bit more, to learn how to learn.

B. Course Learning Outcomes (CLO) covered

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

CLO 1: Demonstrate ability to write, test, and debug code within a development environment.

CLO 2: Produce solutions to computing problems using C programming language in consideration of algorithms within the constraints of a C language's syntax and semantics.

CLO 3: Propose solutions to common safe programming practice.

CLO4: Practice C programming in an ethical manner i.e. without causing software bugs.

Extra: think about how to support personalized learning by exploring C programming techniques.

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 This includes incorporating published and unpublished material, whether in manuscript, printed or electronic form into your work without acknowledging the source (the person and the work).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 assessments submitted must be the student's own work, without any materials generated by AI tools, including direct copying and pasting of text or paraphrasing. Any form of academic misconduct, including using prohibited materials or inappropriate assistance, is a serious offense and will result in a zero mark for the entire assessment or part of it. If there is more than one guilty party, such as in case of collusion, all parties involved will receive the same penalty.

D. Instruction to Students

This assignment is an **individual** assignment. Each student should submit a compressed file (zip/rar) with the name of "YourStudentID.zip" including all c source code files (.c file, .h file), an executable file (.exe file), knowledge point files (.txt file), tracking log files (.txt file), course cumulative learning time file (.dat binary file), a report (Maximum 10 pages excluding the acknowledgment and appendix, font: "Times New Roman", size: 12), and a link of recorded video of the oral presentation. The maximum presentation length is 10 minutes for each student.

An oral presentation should introduce all the materials in the report with consistency. The oral presentation should not exceed a total time of TEN (10) minutes.

The deadline for this assignment is **23:59, 31th Dec. 2023**. Overdue penalty will be given to the assignment that is submitted after the deadline.

Submission type: Softcopy

Submission due date: 31th Dec. 2023

E. Evaluation Breakdown

No.	Component Title	Percentage (%)
1.	Compiling of Knowledge Units (course CST103)	20
2.	Source code	10
3.	Presentation (report)	5
4.	Presentation (oral)	5
	TOTAL	40

F. Task(s)

This task is to design and develop a software tool, called as “knowledge point learning tracking system “(KPLTS for short), which can help student learn C programming (and other courses in the future) more efficiently and individually.

1. According to the course content outline of CST103, as an example of any specific course, the course content outline is divided into chapters, each of which corresponds to a set of CLOs, learning times under different learning methods and total student learning time spent for this chapter. Each chapter is then divided into several sections, and each section is divided into several knowledge units, or knowledge points in pedagogy jargon. Each section or each knowledge unit should have corresponding set of CLOs, learning times under different learning methods and total student learning time spent for this section or knowledge unit (this information is not illustrated in the official course description file). Course, chapter, section and knowledge unit, these four levels of object have certain coherent relationships. Student should design proper data forms to represent these objects in C programming language (for simplicity, student can ignore the CLO column if he/she feels puzzled). (CLO1)

Course Content Outline	CLO*	Teaching and Learning Activities					SLT
		Guided Learning (F2F)				Independent Learning (NF2F)	
		L	T	P	O		
1. A Tutorial Introduction (A Tutorial Introduction, Getting Started, Variables and Arithmetic Expressions, The for statement, Symbolic Constants, Character Input and Output, Arrays, Functions, Arguments - Call by Value, Character Arrays, External Variables and Scope)	1	3	1	1		5	10
2. Types, Operators and Expressions (Variable Names, Data Types and Sizes, Constants, Declarations, Arithmetic Operators, Relational and Logical Operators, Type Conversions, Increment and Decrement Operators, Bitwise Operators, Assignment Operators and Expressions, Conditional Expressions, Precedence and Order of Evaluation)	1,4,2	3	2	2		7	14
3. Control Flow (Statements and Blocks, If-Else, Else-If, Switch, Loops - While and For, Loops - Do-While, Break and Continue, Goto and labels)	1, 4	4	2	2		8	16
4. Functions and Program Structure (Basics of Functions, Functions Returning Non-integers, External Variables, Scope Rules, Header Files, Static Variables, Register Variables, Block Structure, Initialization, Recursion, The C Preprocessor)	1,3	5	3	3		11	22
5. Pointers and Arrays (Pointers and Addresses, Pointers and Function Arguments, Pointers and Arrays, Address Arithmetic, Character Pointers and Functions, Pointer Arrays; Pointers to Pointers, Multi-dimensional Arrays, Initialization of Pointer Arrays, Pointers vs. Multi-dimensional Arrays, Command-line Arguments, Pointers to Functions, Complicated Declarations)	1,3	5	3	3		11	22
6. Structures (Basics of Structures, Structures and Functions, Arrays of Structures, Pointers to Structures, Self-referential Structures, Table Lookup, Typedef, Unions, Bit-fields)	1,3	5	2	2		9	18
7. Input and Output (Standard Input and Output, Formatted Output - printf, Variable-length Argument Lists, Formatted Input - scanf, File Access, Error Handling - Stderr and Exit, Line Input and Output, Miscellaneous Functions)	1,3	5	2	2		9	18
Total		30	15	15	0	60	120

2. Student starts to review the course according to this course content outline and write down all the covered knowledge units, each in a separated text file. Each section should contain at least one knowledge unit. Student can search for the required knowledge units from textbook or Internet and compile a human-readable, friendly looking text file for each knowledge unit. Student may refer to the contents.txt file for the more detailed course outline, in which the minimum number of knowledge units is indicated by the number in the parentheses following each section. (Extra outcome)

3. Student is requested to design and develop a software to help learn this course. The basic requirements are as the following: (1) This software should have a robust menu system to support friendly interactions between the user and the program. (2) User can browse all the compiled knowledge units. (3) User can select certain knowledge unit to start learning. The program will automatically count time and calculate the elapsed learning time after the user ends learning. (4) There should be a human-readable log text file to record all the learning activities, including at least the starting time, ending time, elapsed time, knowledge unit name, student ID and etc. (5) The software should provide a way to show different kinds of cumulative learning times spent by specific student for specific course on specific chapter, section and knowledge unit. (6) There should be a binary file to store all these different kinds of cumulative learning times, which is updated after each learning activity. (CLO1, CLO2, CLO3, CLO4)

4. Student will write a report about this software. The report should cover the initial goals, primary design ideas, user view functionalities and usage, programmer view software structures and modules, any valuable findings got during this programming process, test cases, conclusion and future work. (CLO4)
5. Student will give a presentation, with some slides and demonstration of this software. The maximum presentation length is 10 minutes for each student. (CLO3)

MARKING RUBRICS

Component Title	Creation of knowledge points of course CST103					Percentage (%)	20
Criteria	Score and Descriptors					Weight (%)	Marks
	Excellent (16-20)	Good (12-15)	Average (7-11)	Need Improvement (3-6)	Poor (1-2)		
	1. it covers 100% of all the knowledge points required in the course outline of CST103 2. Each knowledge point file is written in clear and concise English with a good layout. 3. The knowledge point tree is well organized and contains the KB code and the file name of each knowledge point.	1. it covers not less than 80% of all the knowledge points required in the course outline of CST103 2. Each knowledge point file is written in English with a good layout. 3. The knowledge point tree is well organized and contains the KB code and the file name of each knowledge point.	1. it covers not less than 60% of all the knowledge points required in the course outline of CST103 2. Each knowledge point file is written in clear and concise English with a fair layout. 3. The knowledge point tree is well organized and contains the KB code and the file name of each knowledge point.	1. it covers not less than 40% of all the knowledge points required in the course outline of CST103 2. Each knowledge point file is written in clear and concise English with a fair layout. 3. The knowledge point tree is well organized and contains the KB code and the file name of each knowledge point.	1. it covers less than 40% of all the knowledge points required in the course outline of CST103 2. Each knowledge point file is written in clear and concise English with a fair layout. 3. The knowledge point tree is well organized and contains the KB code and the file name of each knowledge point.	20	
TOTAL						20	

Component Title	Source code					Percentage (%)	10
Criteria	Score and Descriptors					Weight (%)	Marks
	Excellent (9-10)	Good (6-8)	Average (4-5)	Need Improvement (2-3)	Poor (0-1)		
	Pass all the test cases without software bugs.	Pass not less than 80% of the test cases.	Pass not less than 50% of all the test cases.	Code seems complete, but with few compile errors.	Code is not complete and with lots of compile errors.	10	
TOTAL						10	
Component Title	Presentation – report					Percentage (%)	5
Criteria	Score and Descriptors						

	Excellent (5)	Good (4)	Average (3)	Need Improvement (2)	Poor (0-1)	Weight (%)	Marks
	Information is presented in effective order. Excellent structure of paragraphs and transitions enhances readability and comprehension.	Information is logically ordered with paragraphs and transitions.	Include vague information.	Details and examples are not organized, and hard to follow and understand.	Details are not related and badly written and hard to follow.	5	
TOTAL						5	

Component Title	Presentation – Oral					Percentage (%)	5
Criteria	Score and Descriptors					Weight (%)	Marks
	Excellent (5)	Good (4)	Average (2-3)		Poor (0-1)		
	*delivers clear opening and closing remarks that capture the attention of the audience and set the mood * provides a “road map” for the audience ▪ each segment relates to the others according to a carefully planned framework	*displays introductory or closing remarks, but segments of the body of the presentation are not presented in a coherent manner. presents the segments of the body of the presentation in a coherent manner, but introductory or closing remarks are missing.	▪ displays some level of organization with discernible theme, but the presentation is not organized clearly or in a coherent manner. introductory and closing remarks are missing.		▪displays neither clear introductory nor closing remarks ▪does not present the segments of the body of the presentation in a coherent manner	5	
TOTAL						5	

Note to students: Please include the marking rubric when submitting your coursework.

ATTENTION!

- Lecturers shall fill-in Section A, B, D, E, F and G
- SECTION A: Brief introduction about the assessment
- SECTION B: Course Learning Outcomes to be covered by this assessment
- SECTION D: Instruction for students on the formation of group, submission type (hardcopy or softcopy), submission due date, etc...
- SECTION E: Evaluation breakdown for the assessment. Example: Project 50% comprises of Report 40% and Presentation 10%
- SECTION F: Question(s) of the assessment
- APPENDIX 1: Marking Rubrics
 - Students shall include the marking rubrics with the coursework submission to lecturers.
 - Lecturers shall mark the students' work according to the marking rubrics as designed.
 - Lecturers shall add the table of marking rubrics according to the number of components as mentioned in SECTION E.
 - Lecturers shall modify the table of marking rubrics accordingly, including the format/ template, column/ row, etc.
- Please add or delete the row in each table if the row not enough or empty.
- If the assignment is in the form of question and answer, for example mathematics courses, lecturer shall delete relevant section(s) but lecturer will need to prepare the appropriate marking scheme.
- Don't forget to delete these red-coloured paragraphs as well in the completed form.