MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية			
Module Title	**Advanced Programming**	**Module Delivery**	
Module Type	Core	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TUCS		
ECTS Credits	8		
SWL (hr/sem)	**200**		
Module Level	1	**Semester of Delivery**	2nd
Administering Department	Computer Science	**College**	CCSM
Module Leader	Mohanad Hatem Ramadhan	**e-mail**	Mohanad.H.Ramadhan@tu.edu.iq
Module Leader's Acad. Title	Assistant Lecturer	**Module Leader's Qualification**	master
Module Tutor	Yahya Laith Khalil	**e-mail**	
Peer Reviewer Name	Mohamed Aktham	**e-mail**	
Scientific Committee Approval Date	07/06/2023	**Version Number**	1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Programming Fundamentals	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. Understanding Advanced Data Structures: The module aims to provide students with a deep understanding of arrays, strings, and their manipulation techniques. Students will learn about multidimensional arrays, character arrays, and string handling functions. 2. Mastery of Pointers: The module aims to develop students' proficiency in using pointers in C++. Students will learn the concepts of memory addresses, pointer arithmetic, and dynamic memory allocation. They will understand how to manipulate data using pointers and how to utilize them for efficient memory management. 3. File Handling and Input/Output Operations: The module aims to introduce students to file handling concepts and techniques in C++. Students will learn how to read from and write to files, open and close files, handle file errors, and perform various input/output operations using file streams. They will understand file modes, buffering, and error handling. 4. File Management and Organization: The module aims to teach students how to manage and organize files effectively in C++. They will learn to create, modify, and delete files, organize file directories, and handle file-related operations. Students will understand the importance of file management in real-world programming scenarios. 5. Practical Application and Problem-Solving: Throughout the module, students will be exposed to practical programming exercises and problem-solving tasks. They will apply the concepts learned to solve real-world programming challenges, consolidating their understanding and enhancing their problem-solving skills. By focusing on arrays, strings, pointers, and file handling in C++, this advanced programming module aims to provide students with a comprehensive understanding of these concepts and their practical application. Students will develop the skills necessary to manipulate complex data structures, handle files, and write efficient and reliable code.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Demonstrate an in-depth understanding of arrays, strings, pointers, and file handling concepts in C++. 2. Apply advanced array operations, such as searching and sorting algorithms, and multidimensional arrays to solve programming problems. 3. Manipulate strings effectively, including concatenation, substring extraction, searching, and sorting. 4. Utilize pointers proficiently for data manipulation, including memory addresses, and

	pointer arithmetic 5. Read from and write to files, perform input/output operations, and handle file-related errors using file streams in C++. 6. Manage and organize files effectively, including creating, modifying, deleting, and organizing file directories. 7. Apply efficient programming techniques, optimize code, and adhere to best practices for writing clean and readable code. 8. Demonstrate problem-solving skills by applying the learned concepts to solve real-world programming challenges. 9. Work collaboratively in teams, communicate effectively, and share knowledge and ideas related to advanced programming concepts. 10. Adapt to new programming concepts and technologies beyond the scope of the course, building a foundation for lifelong learning in programming. These learning outcomes reflect the knowledge, skills, and competencies that students will acquire upon completing the advanced programming course. The outcomes emphasize both theoretical understanding and practical application, preparing students for real-world programming challenges and further studies in the field of computer science.
Indicative Contents المحتويات الإرشادية	1. Review of Basic Programming Concepts: - Recap of fundamental programming concepts, including variables, data types, control structures, and functions. 2. Arrays: - Multidimensional arrays - Array manipulation techniques - Searching and sorting algorithms 3. Strings: - String manipulation and operations - String handling functions 4. Pointers: - Introduction to pointers and their usage - Memory addresses and pointer arithmetic - Pointers to arrays 5. Files: - File handling concepts - Reading from and writing to files - File organization and management

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain programming concepts, C++ syntax, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of programming concepts and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using C++. 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for

	improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	93	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	107	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	7.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	5-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Recap of fundamental programming concepts, including variables, data types, control structures, and functions.
Week 2	Introduction to Arrays (Linear arrays)
Week 3	Searching and Sorting Linear Arrays
Week 4	Multidimensional Arrays and Square Arrays

Week 5	Multiplication of Two Arrays and Re-write TicTacToe game with Arrays
Week 6	Introduction to String and Its Operations
Week 7	More Examples on String
Week 8	Introduction to Pointers
Week 9	Pointer to Array and Pointer Arthmetic
Week 10	First Project Due (Reviewing and Comments)
Week 11	Introduction to Files and Directories
Week 12	Working with Text Files (Read, Write)
Week 13	Working with Binary Files
Week 14	Second Project Due (Students Presentations part1)
Week 15	Second Project Due (Students Presentations part1)

Delivery Plan (Weekly Lab. Syllabus):

المنهاج الاسبوعي للمختبر:

Week No.	**Material Covered**
Week 1	**Getting used to CLI Interfaces and practicing some commands on PowerShell**
Week 2	**Running Examples on Array**
Week 3	**Practicing Arrays further (Searching)**
Week 4	Practicing Arrays further (Sorting)
Week 5	**Running Examples on 2D and Square Arrays**
Week 6	**Running Examples on Strings**
Week 7	**Searching in String**
Week 8	**Running Characters Frequency Example**
Week 9	**Running Examples on Pointers**
Week 10	**Running More Examples on Pointers**
Week 11	**Running Examples on Directories and Files**
Week 12	**Running More Examples on Files**
Week 13	**Running More Advanced Programs on Files**
Week 14	**Wrapping up**
Week 15	**Answering Students Questions and Extra Advising on Real World Application Programming**

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Stroustrup, Bjarne - Programming_ principles and practice using C++-Addison-Wesley (2015)	Yes
Recommended Texts	Olsson, Mikael - C++20 Quick syntax reference: a pocket guide to the language, apis, and library	No
Websites		

Grading Scheme مخطط الدرجات				
Group	**Grade**	التقدير	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.