MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	Discrete Structures			Module Delivery	
Module Type	Basic			☒ Theory ☐ Lecture ☐ Lab ☒ Tutorial ☐ Practical ☐ Seminar	
Module Code	CIT1225				
ECTS Credits	4				
SWL (hr/sem)	100				
Module Level	1		Semester of Delivery	2	
Administering Department	SOFT		College	CIT	
Module Leader	Haneen Talal Jasim Mohammed		e-mail	haneen.talal@uoninevah.edu.iq	
Module Leader's Acad. Title	Lecturer		Module Leader's Qualification	Ph.D.	
Module Tutor			e-mail	E-mail	
Peer Reviewer Name	Name		e-mail	E-mail	
Scientific Committee Approval Date	20/06/2023		Version Number	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives **أهداف المادة الدراسية**	1. Understand the fundamental concepts and methods of discrete mathematics, including propositional logic, predicate logic, set theory, relations, functions, and graphs. 2. Apply discrete mathematics to solve problems in computer science, engineering, and other related disciplines. 3. Develop critical thinking skills for understanding, analyzing, and solving mathematical problems, and communicate effectively using mathematical language. 4. Learn basic proof techniques, including direct proof, proof by contradiction, and proof by induction. 5. Understand algorithmic complexity and how it relates to problems in computer science and engineering. 6. Analyze algorithms using big O notation and evaluate their efficiency. 7. Explore combinatorics, including permutations, combinations, and the binomial theorem. 8. Understand probability theory and its application to discrete structures, including conditional probability and independence. 9. Learn discrete probability distributions, including the Bernoulli, binomial, and Poisson distributions. 10. Explore graph theory and its application to problems in computer science and engineering, including shortest-path algorithms, network flow, and minimum spanning trees.
Module Learning Outcomes **مخرجات التعلم للمادة الدراسية**	1. Define and use the language of propositional and predicate logic to create, analyze, and interpret logical propositions and arguments. 2. Analyze the properties and use of relations, including equivalence relations, partial orderings, and graphs, and apply them to solve problems related to discrete structures. 3. Apply basic principles of set theory, including set operations, cardinality, and functions, to model and analyze problems in computer science and other domains. 4. Develop an understanding of algorithms and their efficiency, including asymptotic notation and algorithm complexity analysis. 5. Demonstrate knowledge of combinatorial techniques and their applications, including permutations, combinations, binomial coefficients, and graph coloring. 6. Analyze the structure and properties of graphs, including trees, paths, connectivity, and cycles, and apply graph theory algorithms to problems in computer science and other domains. 7. Apply counting principles and probability theory to analyze problems in computer science, including the analysis of random algorithms and probabilistic models. 8. Develop an understanding of number theory and its applications to cryptography and coding theory. 9. Design, develop, and analyze algorithms that use different data structures, including arrays, linked lists, stacks, and queues. 10. Apply discrete mathematical techniques to analyze problems in computer science and other domains, including database theory, scheduling, and optimization. 11. Analyze and interpret the behavior of recursive algorithms and use them to solve problems in computer science and other domains. 12. Express mathematical arguments rigorously and formally using

	appropriate mathematical notation and language. 13. Develop an appreciation for the importance of discrete structures and their applications in computer science, engineering, and other domains. 14. Communicate effectively about discrete mathematical concepts and their applications to both technical and non-technical audiences.
Indicative Contents **المحتويات الإرشادية**	1. Propositional and predicate logic: syntax and semantics, truth tables, logical equivalences, normal forms, quantifiers, and logical reasoning.[6 hrs] 2. Set theory: sets, set operations, relations, functions, mappings, cardinality, and applications. [5 hrs] 3. Number theory: divisibility, primes, fundamental theorem of arithmetic, modular arithmetic, and applications to cryptography and coding theory.[4 hrs] 4. Combinatorics: permutations, combinations, binomial theorem, inclusion-exclusion principle, generating functions, and applications.[7 hrs] 5. Graph theory: basics of graph theory, trees, connectivity, planar graphs, coloring, and algorithms.[4 hrs] 6. Algorithms: correctness, efficiency, big-O notation, complexity, recursion, and searching/sorting algorithms.[5 hrs] 7. Discrete probability theory: basic probability concepts, random variables, expected values, probability distributions, and applications.[7 hrs] 8. Recursion: recursive functions, recursive data structures, recursion trees, and solving recurrences.[6 hrs] 9. Formal languages and automata theory: concepts of grammars, regular languages, finite automata, pushdown automata, and Turing machines.[5 hrs] 10. Applications of discrete structures in computer science: database theory, computer networks, optimization, and scheduling.[4 hrs]

Learning and Teaching Strategies **استراتيجيات التعلم والتعليم**	
Strategies	1. Active Learning: Encourage active learning by incorporating group projects, class discussions, and self-assessment activities. This approach helps students to engage with the material and take ownership of their learning. 2. Real-World Examples: Incorporate real-world examples into your lessons to help students understand the relevance of discrete structures in solving practical problems. 3. Visual Aids: Use visual aids, such as diagrams, charts, and graphs, to illustrate complex concepts and help students to visualize abstract ideas. 4. Practice Problems: Provide practice problems that reinforce key concepts and techniques. This approach helps students to build their problem-solving skills and to identify areas where they may need additional help. 5. Scaffolding: When introducing new concepts, use a scaffolding approach in which you break down complex ideas into smaller, more manageable parts. This approach helps students to build a solid foundation of knowledge before moving on to more challenging material. 6. Feedback: Provide timely and constructive feedback on student work. This approach helps students to identify where they are excelling and where they need to improve.

	7. Multiple Modes of Learning: Incorporate multiple modes of learning, such as lectures, readings, and online resources. This approach accommodates different learning styles and helps to reinforce key concepts. 8. Personalization: Personalize your teaching to meet the needs of individual students. This approach helps students to feel supported and engaged, which in turn fosters deep learning and mastery of the material.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	47	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	3
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	53	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	100		

Module Evaluation تقييم المادة الدراسية					
As		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	3	30% (30)	3, 5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report				
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week	**Material Covered**
Week 1	Introduction to Discrete Structures
Week 2	Number Theory
Week 3	Counting Principles
Week 4	Graph Theory
Week 5	Trees
Week 6	Boolean Algebra
Week 7	Predicate Logic
Week 8	Induction and Recursion
Week 9	Sets and Relations
Week 10	Functions
Week 11	Combinatorial Analysis
Week 12	Algorithm Analysis
Week 13	Cryptography
Week 14	Complexity Theory
Week 15	Review and Final Exam
Week 16	**Preparatory week before the final Exam**

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	1. "Discrete Mathematics and Its Applications" by Kenneth H. Rosen - This is a popular and widely used textbook that covers all the essential topics with clear explanations, examples, and exercises. 2. "Discrete Mathematics: An Open Introduction" by Oscar Levin - This book is available for free online and provides a comprehensive introduction to the subject with an emphasis on computer science applications.	
Recommended Texts	1. "Concrete Mathematics: A Foundation for Computer Science" by Ronald L. Graham, Donald E. Knuth, and Oren Patashnik - This book is a classic and uses a unique blend of combinatorics and analysis to introduce the concepts of discrete mathematics. 2. "Discrete Mathematics and Its Applications with Combinatorics and Graph Theory" by Kenneth H. Rosen and John G. Michaels - This textbook provides a comprehensive introduction to discrete mathematics, combinatorics, and graph theory, with a strong focus on algorithms and applications.	
Websites	Internet , youTube	

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.