MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	Computation Theory			**Module Delivery**	
Module Type	Core			☒**Theory** ☒**Lecture** ☐**Lab** ☐**Tutorial** ☐**Practical** ☐**Seminar**	
Module Code	**MU010602103**				
ECTS Credits	4				
SWL (hr/sem)	**100**				
Module Level	UGII		**Semester of Delivery**	3	
Administering Department	Computer Science		**College**	College of Science	
Module Leader	Dr. Zied O. Ahmed		**e-mail**	zied_othman@uomustansiriyah.edu.iq	
Module Leader's Acad. Title	**Assistant Prof.**		**Module Leader's Qualification**		
Module Tutor	Dr. Hassan Kassim Mohammad		**e-mail**	hassan@uomustansiriyah.edu.iq	
Peer Reviewer Name	Dr. Zied O. Ahmed		**e-mail**	zied_othman@uomustansiriyah.edu.iq	
Scientific Committee Approval Date	22 /4 /2024		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Discrete Mathematics (I)	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	**This course presents the:** Theory of computation is the theoretical study of capabilities and limitations of Computers (Theoretical models of computation). To introduce students to the mathematical infrastructure of computation and specifically to automata theory, the theory of formal languages and grammars, the concepts of algorithm, diagnosability, complexity and computability.

	To develop students' ability to understand and construct mathematical proofs for calculations and algorithms.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	**Upon completion of the course, the students will be able to:** 1- an understanding of basic concepts in the theory of computation through simple models of computational devices. 2- apply models in practice to solving problems in diverse areas such as string searching, pattern matching, cryptography, and language design; 3- understand the limitations of computing, the relative power of formal languages and the inherent complexity of many computational problems.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. **UNIT 1 :** [8 hrs] Introduction, terminology, definitions, Sets and operations, languages, Regular Expressions RE **UNIT 2** : [22 hrs] Finite Automata FA, Deterministic Finite Automaton DFA, Moore and Mealy machines, Kleene's Theorem **UNIT 3** : [28 hrs] Context-free Languages, Pushdown Automata, CFG derivation, trees, Parsing, Chomsky normal form, Ambiguous CFL's **UNIT 4** : [15 hrs] PDA to CFG/CFL, TURING MACHINES TM

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	**Learning Strategies:** The best way to learn and remember Computation Theory is to practice them and draw the automata using UML tools. You can also find books, courses, or tutorials that offer exercises and examples. 1- Hands-on Practice: Programming is a practical skill that requires practice. Allocate dedicated time to work on Computation theory exercises, small projects, and problem-solving activities. 2- Reading types of Computation theory: Study and analyze existing Computation theory examples and programs written by experienced developers. write the programs to enhance your understanding of Computation theory. 3- Collaborative Learning: Engage in discussions and collaborations with fellow learners or join programmer communities and forums to share knowledge, ask questions, and solve problems collectively.

	Teaching Strategies: 1- Conceptual Explanation: Start by providing clear and concise explanations of Computation Theory concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2- Hands-on Coding Exercises: Include practical Computation Theory exercises and assignments that reinforce the concepts taught. Convert the types of Computation Theory to programs. Encourage students to apply the algorithm to practice, helping them develop their coding skills. 3- Interactive Sessions: Conduct live sessions where students can actively participate, ask questions, and solve problems alongside the instructor. This approach promotes engagement and practical learning. In addition to the practical project 4- Resources and References: Share relevant Computation Theory resources, recommended textbooks, and online tutorials to supplement the curriculum and encourage further exploration.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	63	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	37	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	2
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	**100**		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	2	10% (10)	6 and 10	LO #1, #2
	Assignments	2	8% (8)	4 and 12	LO #1, #2, #3
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3
Summative assessment	**Midterm Exam**	2hr	10% (10)	8	LO #1, #2, #3
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction, terminology, definitions
Week 2	Sets and operations
Week 3	languages
Week 4	Regular Expressions RE
Week 5	Finite Automata FA
Week 6	Deterministic Finite Automaton DFA
Week 7	Moore and Mealy machines
Week 8	Kleene's Theorem
Week 9	Context-free Languages
Week 10	Pushdown Automata
Week 11	CFG derivation trees Parsing
Week 12	Chomsky normal form
Week 13	Ambiguous CFL's
Week 14	PDA to CFG/CFL
Week 15	TURING MACHINES TM

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	
Week 2	
Week 3	
Week 4	
Week 5	
Week 6	
Week 7	
Week 8	
Week 9	
Week 10	
Week 11	

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	**[1]: Introduction to Computer Theory 2nd Edition Daniel I. A. Cohen John Wiley & Sons, Inc 1997. ISBN 0-471-13772-3**	Yes
Recommended Texts	**[1]: Introduction to Automata Theory, Languages, and Computation, 2/E, John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, Addison-Wesley 2001. ISBN 0-201-44124-1.**	Yes
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.