

# MODULE DESCRIPTION FORM

## نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Computation theory		Module Delivery
Module Type	Supportive		<input checked="" type="checkbox"/> Theory <input checked="" type="checkbox"/> Lecture <input type="checkbox"/> Lab <input checked="" type="checkbox"/> Tutorial <input type="checkbox"/> Practical <input type="checkbox"/> Seminar
Module Code	TUCS112		
ECTS Credits	6		
SWL (hr/sem)	150		
Module Level	1	Semester of Delivery	
Administering Department	Computer Science	College	CCSM
Module Leader	Marwa Adeeb Mohammad	e-mail	Marwa.a.aljawaherry@tu.edu.iq
Module Leader's Acad. Title	Assistant Professor	Module Leader's Qualification	master
Module Tutor		e-mail	
Peer Reviewer Name		e-mail	
Scientific Committee Approval Date	01/01/2024	Version Number	1.0

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

## Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

<p><b>Module Aims</b> أهداف المادة الدراسية</p>	<p>Theory of computation is the theoretical study of capabilities and limitations of Computers (Theoretical models of computation). Providing students with:</p> <ol style="list-style-type: none"><li>1. Understand of basic concepts in the theory of computation through simple models of computational devices.</li><li>2. Apply models in practice to solving problems in diverse areas such as string searching, pattern matching, cryptography, and language design.</li><li>3. Understand the limitations of computing, the relative power of formal languages and the inherent complexity of many computational problems.</li><li>4. Be familiar with standard tools and notation for formal reasoning about machines and programs.</li></ol>
<p><b>Module Learning Outcomes</b> مخرجات التعلم للمادة الدراسية</p>	<p>Upon successful completion of this module, students should be able to:</p> <p>Cognitive objectives:</p> <ol style="list-style-type: none"><li>1. Enabling the student to know and understand the theoretical principles of the programming process.</li><li>2. Enabling the student to know the intellectual framework of computer basics.</li><li>3. Enabling the student to know the practical applications of the programming foundation that the subject has.</li><li>4. Enabling the student to explain the steps of program implementation.</li></ol> <p>skills objectives for the course:</p> <ol style="list-style-type: none"><li>1. Designing basic diagrams for computational theoretical methods.</li><li>2. Writing algorithms.</li><li>3. Writing and implementing methods.</li><li>4. Following up on the implementation and planning of algorithms and discovering errors.</li></ol> <p>Teaching and learning methods, theoretical lectures, using projectors for clarification with illustrative examples, drawing diagrams that facilitate the process of understanding and conveying the idea to the student.</p>
<p><b>Indicative Contents</b> المحتويات الإرشادية</p>	<p>Focuses interest on studying how calculations and reasoning are performed in the context of computer systems.</p> <ol style="list-style-type: none"><li>1. Computing Paradigms: Paradigms such as the Turing machine, quantum computing are explored and used to analyze computational capabilities and limits.</li><li>2. Description Languages: Computational theory studies the languages and models that are used to describe and analyze computational operations, such as neural language and conditional language.</li></ol>

	<ol style="list-style-type: none"> <li>3. Computational complexity: It is concerned with analyzing computer complexity and classifying problems according to their computational difficulty and the ability of computer systems to solve them.</li> <li>4. Mental Machines: Computational theory studies the relationship between computing and the human mind, and whether the mind can be represented and simulated by computational machines.</li> <li>5. Artificial Intelligence: Artificial intelligence is an important application of computational theory, where computational models are used to develop artificial intelligence systems capable of thinking and making decisions.</li> </ol>
--	--

<b>Learning and Teaching Strategies</b> استراتيجيات التعلم والتعليم	
<b>Strategies</b>	<ol style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>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++.</li> <li>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.</li> </ol>

<b>Student Workload (SWL)</b> الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
<b>Structured SWL (h/sem)</b> الحمل الدراسي المنتظم للطالب خلال الفصل	77	<b>Structured SWL (h/w)</b> الحمل الدراسي المنتظم للطالب أسبوعيا	5.1
<b>Unstructured SWL (h/sem)</b> الحمل الدراسي غير المنتظم للطالب خلال الفصل	73	<b>Unstructured SWL (h/w)</b> الحمل الدراسي غير المنتظم للطالب أسبوعيا	4.8
<b>Total SWL (h/sem)</b> الحمل الدراسي الكلي للطالب خلال الفصل	150		

<b>Module Evaluation</b> تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
<b>Formative assessment</b>	<b>Quizzes</b>	2	10% (10)	5, 11	LO #1-4 , LO# 5-7
	<b>Assignments</b>	2	10% (10)	7, 12	LO #1-4 , LO# 5-7
	<b>Projects</b>	1	10% (10)	continuous	LO #1-7
	<b>Report</b>	1	10% (10)	14	LO # 4 - 7
<b>Summative assessment</b>	<b>Midterm Exam</b>	2 hr	10% (10)	11	LO # 1-6
	<b>Final Exam</b>	2 hr	50% (50)	16	All
<b>Total assessment</b>			100% (100 Marks)		

<b>Delivery Plan (Weekly Syllabus)</b> المنهاج الاسبوعي النظري	
Week No.	Material Covered
<b>Week 1</b>	Set, string, alphabet and language
<b>Week 2</b>	Regular expression
<b>Week 3</b>	Finite state automata, deterministic and nondeterministic finite state automata.
<b>Week 4</b>	Equivalence between deterministic and nondeterministic finite state automata
<b>Week 5</b>	Finite state automata with empty move.
<b>Week 6</b>	Equivalence of NFA with and without $\epsilon$ move
<b>Week 7</b>	The equivalence between Moore and Mealy machine
<b>Week 8</b>	Grammar, Chomsky hierarchy of languages.
<b>Week 9</b>	- The regular grammars and regular languages.
<b>Week 10</b>	- Closure properties of regular sets (union, concatenation and kleen closure).
<b>Week 11</b>	Regular expression, closure properties of regular languages (intersection, complementation

	and substitution)
<b>Week 12</b>	Decision procedures for regular sets(emptiness, finiteness, containment and equivalence).
<b>Week 13</b>	Context- free grammars and languages with their Properties.
<b>Week 14</b>	Context-free grammar without empty string production ( $\lambda$ -free grammar).
<b>Week 15</b>	Derivation trees.

### Learning and Teaching Resources

#### مصادر التعلم والتدريس

	Text	Available in the Library?
<b>Required Texts</b>	Introduction to Computer Theory 3 <sup>rd</sup> Edition michael sipser, USA, 2013. ISBN-13: 978-1-133-18779-0.	Yes
<b>Recommended Texts</b>	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.	No
<b>Websites</b>		

### Grading Scheme

#### مخطط الدرجات

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