	Ministry of Higher Education and Scientific Research - Iraq Al-Nahrain University College of Science Computer Science Department	

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	Language Translators			**Module Delivery**	
Module Type	Core			☒ **Theory** ☐ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	Comp3219				
ECTS Credits	5				
SWL (hr/sem)	60				
Module Level	3		**Semester of Delivery**	3	
Administering Department	Computer Science		**College**	Science	
Module Leader	Dr. Bahera Hani Nayef		**e-mail**	Bahera.hani@nahrainuniv.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	Ph.D.	
Module Tutor	L.Ass. Ruaa Abdullah Jaber		**e-mail**	ruaa.abdullah@nahrainuniv.edu.iq	
Peer Reviewer Name	L. Ass. Ibtihal Abbas		**e-mail**	ibtihal.abbas@nahrainuniv.edu.iq	
Scientific Committee Approval Date	01/02/2026		**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 Aims **أهداف المادة الدراسية**	**Main Objectives of Translator Training Programs** 1. Advanced Linguistic & Transfer Competence 2. Specialized Domain Knowledge 3. Technical Tool & Technology Proficiency 4. Research Skills & Professional Ethics 5. Career Preparation & Business Acumen
Module Learning Outcomes **مخرجات التعلم للمادة الدراسية**	Here are 4 concise module learning outcomes for a Language Translators subject: ● Explain the concepts and types of language translators and the phases of a compiler. ● Apply lexical and syntax analysis techniques to analyze source programs. ● Construct parse trees and symbol tables for simple programming languages. ● Design and evaluate basic compiler components and parsing strategies. **Critical thinking skills** Here are concise critical thinking skills specifically for Language Translators (LT): ● Analyze language syntax and semantics to identify ambiguities and translation errors. ● Evaluate different parsing and translation strategies for correctness and efficiency. ● Diagnose and debug errors across compiler phases using logical reasoning. ● Optimize translation processes by selecting appropriate data structures and algorithms. **Creativity** Designing innovative grammar rules and language structures • Creating efficient lexical, syntax, and semantic analysis solutions • Developing optimized intermediate code and target code generation methods • Adapting and extending compiler techniques to support new language features **Communication skills** • Clearly explaining compiler concepts, phases, and translation processes • Documenting grammar rules, algorithms, and design decisions effectively • Communicating technical ideas using diagrams, parse trees, and examples • Collaborating with team members to design and debug translator components **Teamwork skills** • Active Listening & Clear Inquiry: Accurately grasp requirements and ask precise questions to confirm understanding. • Clarity & Concise: Communicate updates, issues, and decisions

	clearly and promptly. • Diplomacy & Tact: Handle feedback and disagreements professionally to maintain collaboration. • Cultural Sensitivity**:** Work effectively with diverse global teams and clients
Indicative Contents **المحتويات الإرشادية**	Indicative content includes the following. ● **Introduction to programming languages** ● Compiler and interpreter ● Compiler stages ● Study of the Lexical Stage Simulating DFA (Methods (Functions) ● NFA to DFA ● Top-Down Parsing ● Recursive Descent Parsing ● Ambiguous Grammar ● Left Factoring ● Eliminating Left Recursion

Learning and Teaching Strategies **استراتيجيات التعلم والتعليم**	
Strategies	• **Immersive Reading & Parallel Text Analysis:** Critically reading content in both languages and studying professional translations to internalize style and technique. • **Deliberate Practice with Feedback:** Actively translating challenging texts and seeking critique to improve decision-making. • **Specialization & Tool Mastery:** Systematically building glossaries in chosen fields and practicing with translation technology (CAT tools, etc.). • **Problem-Based Learning:** Tackling authentic translation challenges that mimic real-world projects. **Core Teaching Strategies (For the Educator/Trainer)** • **Project-Based Learning (PBL):** Running simulated, end-to-end translation projects to teach workflow, teamwork, and tools. • **Think-Aloud Protocols (TAP):** Having students verbalize their translation process to expose and refine decision-making. • **Error Analysis & Commentary:** Having students compare translations and defend their choices in writing to link theory to practice. • **Flipped Classroom & Role-Playing:** Using class time for hands-on problem-solving and simulating professional scenarios (client meetings, revisions).

Student Workload (SWL) **الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا**

Structured SWL (h/sem) **الحمل الدراسي المنتظم للطالب خلال الفصل**	60	**Structured SWL (h/w)** **الحمل الدراسي المنتظم للطالب أسبوعيا**	4
Unstructured SWL (h/sem) **الحمل الدراسي غير المنتظم للطالب خلال الفصل**	65	**Unstructured SWL (h/w)** **الحمل الدراسي غير المنتظم للطالب أسبوعيا**	4.3
Total SWL (h/sem) **الحمل الدراسي الكلي للطالب خلال الفصل**	125		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)		All
	Assignments	2	10% (10)		All
	Projects / Lab.	1	20% (20)	Continuous	All
	Report	0			
Summative assessment	**Midterm Exam**	2 hr	10% (10)		All
	Final Exam	3hr	50% (50)		All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	Ch1: foundational understanding of programming languages, including theoretical concepts and practical applications. Theoretical elements include language components and classification, HW1. Computer language difference
Week 2	Ch2: introduction to Compiler and interpreter HW2. Comparison between compiler and interpreter
Week 3	Ch2: \Compiler stages
Week 4	Ch2: Symbol Table Management, Error Handling
Week 5	Ch3: Semantic analyzer
Week 6	Ch3: Intermediate code generation and Intermediate code Optimization
Week 7	Mid Term Exam #1
Week 8	Ch4: Study of the Lexical Stage Simulating DFA
Week 9	Ch4: NFA to DFA HW3: convert NAF to DFA
Week 10	Ch5: Minimizing DFA, Simulating NFA,
Week 11	Ch5: Regular Expression to DFA HW4: comparison of simulating NFA and DFA
Week 12	Mid Exam #2
Week 13	Ch6: CFG (Context-Free Grammar) 1-Top-Down Parsing 2-Recursive Descent Parsing
Week 14	Ch6: Ambiguous Grammar, Left Factoring, Eliminating Left Recursion

Week 15	**review**

Delivery Plan (Weekly Lab. Syllabus) **المنهاج الاسبوعي للمختبر**	
	Material Covered
Week 1	Lab 1: Opening and reading a file using the Java language.
Week 2	Lab 2: Read from text file and eliminating spaces and comments
Week 3	Lab 3: Start building a token table.
Week 4	Lab 4: Complete the token table
Week 5	Lab 5: Complete the token table
Week 6	Lab 6: Complete the token table
Week 7	Lab 7: Mid-term Exam
Week 8	Lab 8: Completion of creating the word table, The grouping of phases into passes,
Week 9	Lab 9: Initiating the construction of the symbol table
Week 10	Lab 10: Completion of the symbol table, Building a table for error messages
Week 11	Lab 11: Complete the token table
Week 12	Lab 12: Mid-term Exam
Week 13	Lab 13: Building DFA (Deterministic Finite Automaton) for decimal numbers and floating points.
Week 14	Lab 14: student presentation
Week 15	Lab 15: review

Learning and Teaching Resources **مصادر التعلم والتدريس**		
	Text	**Available in the Library?**
Required Texts	**Compiler (principles, techniques, and tools)** second edition Alfred V.Aho Columbia University Monica S.Lam Stanford University Ravi Sethi Avaya Jeffrey D.Ullman	free online
Recommended Texts	Cooper, Keith D., and Linda Torczon. *Engineering a Compiler / Keith D. Cooper, Linda Torczon.* Third edition.., Morgan Kaufhann Publishers, 2022.	free online
Websites	https://youtu.be/nv9J5Jb7IxM Intermediate code generation (slideshare.net) https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis https://www.javatpoint.com/automata-conversion-from-nfa-to-dfa /	

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.