

وزارة التعليم العالي و البحث العلمي
جهاز الاشراف و التقويم العلمي
دائرة ضمان الجودة و الاعتماد الاكاديمي



الجامعة المستنصرية – كلية العلوم

2025-2024

نظام بولونيا

دليل وصف البرنامج الاكاديمي و المقرر الدراسي

2025

اسم الجامعة: المستنصرية

الكلية : العلوم القسم العلمي : علوم الحاسوب

اسم البرنامج الاكاديمي او المهني : بكالوريوس علوم الحاسوب

اسم الشهادة النهائية : بكالوريوس علوم الحاسوب

النظام الدراسي : نظام بولونيا

تاريخ اعداد الوصف :

تاريخ ملء الملف :

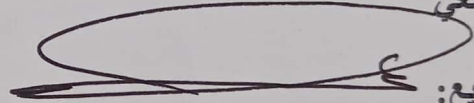


التوقيع :

اسم رئيس القسم : أ.م.د. مصعب رياض عبد الرزاق

التاريخ : 2025 / /

دقق الملف من قبل شعبة ضمان الجودة و الاداء
الجامعي

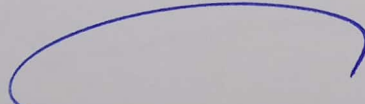


التوقيع :

اسم مدير شعبة ضمان الجودة و الاداء الجامعي :

أ.م.د. علي جاسم الحافظ

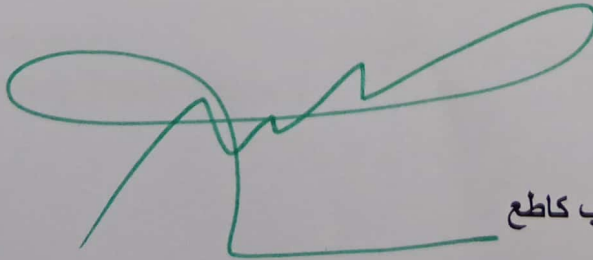
التاريخ : 2025 / /



التوقيع :

اسم معاون القسم : أ.م.د. احمد مهدي رحيمة

التاريخ : 2025 / /



مصادقة السيد العميد : أ.د. ميثاق طالب كاطع

التوقيع :

التاريخ : 2025 / /

وصف البرنامج الأكاديمي

• وصف البرنامج الأكاديمي

يلخص هذا الوصف خصائص البرنامج ونتائج التعلم المتوقعة من الطلاب، ويركز على مدى استفادتهم القصوى من الفرص المتاحة. ويرفق به وصف لكل مقرر دراسي ضمن البرنامج.

المؤسسة التعليمية	الجامعة المستنصرية
الكلية / القسم العلمي	كلية العلوم / قسم علوم الحاسوب
اسم البرنامج الأكاديمي	علوم الحاسوب
اسم الشهادة النهائية	بكالوريوس في علوم الحاسوب
نظام الدراسة	نظام بولونيا
برنامج الاعتماد المعتمد	ABET
تأثيرات خارجية أخرى	لا يوجد

• مخرجات التعلم المطلوبة وأساليب التدريس والتعلم والتقييم

أ. المعرفة والفهم

1. فهم أساسيات الرياضيات المتعلقة بعلوم الحاسوب.
2. معرفة وفهم مجموعة واسعة من المبادئ والأدوات المتاحة لمطوري البرمجيات، مثل منهجيات التصميم، واختيار الخوارزميات، ولغة البرمجة، ومكتبات البرمجيات، وتقنيات واجهة المستخدم.
3. فهم مبادئ مختلف التطبيقات ومجالات البحث الحالية، بما في ذلك الذكاء الاصطناعي، وقواعد البيانات، وهندسة البرمجيات، والشبكات، والأنظمة الموزعة.
4. معرفة وفهم مجموعة واسعة من البرمجيات والأجهزة المستخدمة في تطوير أنظمة الحاسوب.
5. معرفة وفهم المسؤوليات المهنية والأخلاقية لممارسي علوم الحاسوب، بما في ذلك فهم أهمية الجودة والأمان وأخلاقيات الحاسوب

ب. المهارات الفكرية

1. تحليل مجموعة واسعة من المشكلات وتقديم حلول لها باستخدام خوارزميات وهياكل ومخططات مناسبة، وغيرها من الطرق المناسبة.
2. تصميم وتنفيذ نظام برمجي كبير الحجم.
3. تحديد مجموعة من الحلول وتقييمها وتبريرها بشكل نقدي.
4. ممارسة التعلم الذاتي باستخدام الدورات الإلكترونية.

ج. أساليب التدريس والتعلم

1. المحاضرات والمناقشات
2. الواجبات المنزلية والمهام
3. المشاريع والعروض التقديمية (العروض التقديمية)
4. المهارات القابلة للنقل والصفات الشخصية
5. الندوات
6. مشاريع العمل الجماعي

د. أساليب التقييم

1. الاختبارات
2. الامتحانات اليومية
3. العروض

هـ. المهارات المتعلقة بالموضوع

1. تخطيط وتنفيذ مشروع فردي/جماعي رئيسي في مجالات علوم الحاسوب.
2. إعداد وتقديم تقارير تقنية شفوية وكتابية متماسكة ومنظمة.
3. تقديم عروض تقديمية تقنية مناسبة للوقت والمكان والجمهور.
4. استخدام المراجع العلمية بفعالية والاستخدام الأمثل لموارد الإنترنت.
5. تصميم برامج الحاسوب وكتابتها وتصحيح أخطائها باللغات المناسبة.
6. استخدام أدوات دعم التصميم الحاسوبية المناسبة.

ملحق هيكل البرنامج رقم (3)

Lab	Theoretical		Module Code	Module Name	Level One
2	3	اساسيات البرمجة 1	MU010601101	Programming Fundamentals (I)	Semester 1
2	3	مهارات الحاسوب 1	MU010601102	Computer Skills (I)	
	3	هياكل متقطعة	MU010601103	Discrete Structure	
	2	اخلاقيات الحاسوب	MU010601104	Computer Ethics	
	2	الديمقراطية و حقوق الانسان	MU02	Democracy and Human Rights	
	2	اللغة العربية	MU06	Arabic Language	

Lab	Theoretical		Module Code	Module Name	Level One
2	3	اساسيات البرمجة 2	MU010601205	Programming Fundamentals (I)	Semester 2
2	3	مهارات الحاسوب 2	MU010601206	Computer Skills (I)	
	3	تصميم منطقي	MU010601207	Logic Design	
	3	الرياضيات	MU010601208	Mathematics	
	3	تكنولوجيا المعلومات	MU010601209	Information Technology	
	2	اللغة الانكليزية	MU01	English	

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Programming Fundamentals (I)		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☒ Tutorial ☐ Practical ☐ Seminar
Module Code	MU010601101		
ECTS Credits	8		
SWL (hr/sem)	200		
Module Level	1	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader		e-mail	
Module Leader's Acad. Title	Assistant. Prof	Module Leader's Qualification	MSc.
Module Tutor		e-mail	
Peer Reviewer Name	Dr. Jamal Nasir Hasoon	e-mail	jamal.hasoon@uomustansiriyah.edu.iq
Scientific Committee Approval Date	01/06/2023	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	This course Introduces the fundamental concepts of structured programming and provides a comprehensive introduction to programming for computer science and technology majors. Topics include software development methodology, data types, control structures, and the mechanics of running, testing, and debugging. This course assumes computer literacy. This course aims to learn the fundamental principle of structured programming by using C# language. this course will be covering the following topics: fundamentals of computer organization, understanding the generation of high-level programming languages, Basic syntax, and semantics of higher-level language. Study and analysis of simple problems solving: Algorithms, Pseudocode, and Flowchart. Variables, types, expressions, and assignments. simple input-output statement. Conditional structures: - (If, if/else, nested if, switch) and iterative control structures: (while do- while and for repletion structure). the mechanics of running, testing, and debugging.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand the basics of computer system, and Describe how data are represented, manipulated, and stored in a computer. 2. Categorize different programming languages and their uses. 3. Understand and use the fundamental concepts of data types, structured programming, algorithm, flowchart and pseudo code. 4. Understanding of algorithms, flowchart and pseudo code, 5. □ Understand the general structure of C# program. 6. With ability to be able to draw flowchart, write algorithms, and write a program using C# language. 7. With ability to be able to write a program using C# language using Conditional control structures: - ((If, if/else, nested if, switch), and iterative control structures: while, do- while and for repletion structure). 8. Apply computer programming concepts to problems or situations, and demonstrate appropriate design, coding, testing, and documenting of computer programs that solve these problems or situations.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. Introduction to Programming: [14 hrs] ○ What is programming? ○ Programming languages and paradigms. ○ The role of programming in software development. ○ Study analysis of Simple Problem Algorithms, Pseudo code, and flowchart. 2. Variables and Data Types: [10 hrs] ○ Variables and their types. ○ Basic data types (e.g., integers, floats, strings).

	○ Type conversion and casting. ○ Variable assignment and naming conventions. 3. Operators and Expressions: [10 hrs] ○ Arithmetic operators (+, -, *, /, %). ○ Relational operators (==, !=, >, <, >=, <=). ○ Logical operators (&&, , !). ○ Bitwise operators (&, , ^, <<, >>). ○ Operator precedence and associativity. 4. Control Flow: [54 hrs] ○ Conditional statements (if, else if, else). ○ Switch statements. ○ Looping structures (for, while, do-while). ○ Control flow manipulation (break, continue). ○ Nesting control structures.
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning Strategies: 1. Hands-on Practice: Programming is a practical skill that requires practice. Allocate dedicated time to work on coding exercises, small projects, and problem-solving activities. 2. Code Reading and Analysis: Study and analyze existing code examples and programs written by experienced developers. Understand the logic, structure, and flow of the code to enhance your understanding of programming concepts. 3. Break Problems into Smaller Parts: When faced with complex programming problems, break them down into smaller, manageable tasks. This approach, known as "divide and conquer," helps tackle the problem systematically. 4. Collaborative Learning: Engage in discussions and collaborations with fellow learners or join programming communities and forums to share knowledge, ask questions, and solve problems collectively. 5. Debugging and Troubleshooting: Debugging is an essential skill in programming. Practice identifying and fixing errors in your code, as it enhances your problem-solving abilities and helps you understand the logic behind the code. Teaching Strategies:

	1. Conceptual Explanation: Start by providing clear and concise explanations of programming concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2. Hands-on Coding Exercises: Include practical coding exercises and assignments that reinforce the concepts taught. Encourage students to apply the theory into practice, helping them develop their coding skills. 3. Visualizations and Diagrams: Use visual aids like diagrams, flowcharts, or pseudocode to illustrate programming concepts and algorithms. Visual representations can aid understanding, especially for beginners. 4. Interactive Coding Sessions: Conduct live coding sessions where students can actively participate, ask questions, and solve problems alongside the instructor. This approach promotes engagement and practical learning. 5. Resources and References: Share relevant programming resources, recommended textbooks, and online tutorials to supplement the curriculum and encourage further exploration.
--	--

Student Workload (SWL)			
الحمل الدراسي للطلاب محسوب ل ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	109	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	91	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	6
Total SWL (h/sem) الحمل الدراسي الكلي للطلاب خلال الفصل	200		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 12	LO #3, #4, #5.#6, #7, #8
	Assignments	2	8% (8)	7 and 13	LO #3, #4, #5.#6, #7, #8
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)	14	LO #3, #4, #5,#6, #7, #8
Summative assessment	Midterm Exam	2hr	10% (10)	10	LO #1, #2, #3, #4, #5,#6,#7

	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to computer system, basic syntax and semantics of higher-level statement language.
Week 2	Study analysis of Simple Problem Algorithms,
Week 3	Pseudo code, and flowchart
Week 4	Arithmetic Operations and Decision Making: Equality and Relational Operators
Week 5	Math Class Methods / Working with Identifiers, Variables, Constants, Keywords
Week 6	Working with Identifiers, Variables, Constants, Keywords
Week 7	control structures: if selection structure
Week 8	If else selection structure.
Week 9	Nested if /else structure.
Week 10	Midterm Exam / Switch Multiple Selection Structure
Week 11	assignment operators/ increment and decrement operators / Control flow manipulation (break, continue).
Week 12	While repetition structure
Week 13	Do/While repetition structure
Week 14	For repetition structure
Week 15	Nesting repetition structures.
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Writing Program simple C# language and how to correct errors and how to implement
Week 2	Writing program in C#
Week 3	Examples (input, output)
Week 4	Examples (input, output)
Week 5	Examples (Math Class)
Week 6	First exam / Examples (if selection structure)

Week 7	Examples (if selection structure)
Week 8	Examples (If /else selection structure.)
Week 9	Examples (Nested if /else structure)
Week 10	Examples (Switch Multiple Selection Structure)
Week 11	Examples (assignment operators/ increment and decrement operators)
Week 12	Examples (While repetition structure)
Week 13	Examples (Do/While repetition structure)
Week 14	Examples (For repetition structure) / examples Nesting repetition structures.
Week 15	examples Nesting repetition structures. / Second Exam

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	- "Programming Basics with C#" Book and Video Lessons Svetlin Nakov and Team eBook 2019 - Visual C# How to Program, 6th edition Published by Pearson (July 14th 2021) - Copyright © 2017	Yes
Recommended Texts	- Deitle and Deitle. "C Sharp How to Program", Prentic Hall, 2009	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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Computer Skills I		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010601102			
ECTS Credits	7			
SWL (hr/sem)	175			
Module Level	1	Semester of Delivery		1
Administering Department	Computer science	College	College of science	
Module Leader	Salah Taha		e-mail	Salah.taha@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Asst. Professor		Module Leader's Qualification	MSc.
Module Tutor			e-mail	
Peer Reviewer Name	Yasamin Hamza		e-mail	yhamza@uomustansiriyah.edu.iq
Scientific Committee Approval Date	01/06/2023		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. To understand computer basics, identify different computers and software types, computers system components. 2. To be able to identify computer parts, choosing appropriate storage devices and be aware about storage problems. 3. Understand processors and different system software 4. Understanding the Windows operating system. 5. Understanding different software. 6. This course deals with Office 2010, identifying features, benefits, and the usage of this version
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	At the end of the module the student will be able to 1. Understanding the computer basics (H.W, S.W, firmware, networking) and how it evolved. 2. Understand how computers communicate. The students will be able to choose the appropriate storage, input and output devices. 3. Understand the processor and choose the right computer specification according to the task required. 4. Format the computer and install windows operating system 5. Start up, shutting down, manage user accounts 6. Using windows system utilities, and trouble shooting common windows problem. 7. Creating folders, files, copy remove and rename files and folders. 8. Managing the computer desktop (arranging folders and files, deleting and restoring files(folders). 9. Understanding different types of business productivity software and how to manage your application. 10. Using basic features of office 2010 applications 11. Saving, opening and rename documents. Changing fonts, size and color of documents. 12. Printing and sharing files. 13. How to prepare reports by combining more than one application
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. Part A - Computer types computer generations, computer types, computer applications, and first computer architecture. [20 hrs] Part B- Computers components Hardware (input and output devices, processor, external memory, internal memory) Software (types of software). [22 hrs] Part C- Maintain the computer Connected the parts of the hardware, Computer formatting, and software installation. [24 hrs]

	Part D- Windows 10 installation of Windows 10, Windows 10 interface types, and work with interfaces. [28 hrs]
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	In this course, students will be urged to participate in the lectures by preparing reports on each topic and presenting them during the lecture in order to increase the information related to the lecture. In addition, Applying computer formatting processes and installing programs in order to gain the necessary experience in this field

Student Workload (SWL) الحمل الدراسي للطلاب محسوب ل ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	94	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	81	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	5
Total SWL (h/sem) الحمل الدراسي الكلي للطلاب خلال الفصل	175		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	3	10% (10)	4,8 and 14	LO #1-4 and LO #6- #8 and LO #9 - #13
	Assignments	2	10% (10)	6 and 10	LO #1- #8 and LO #9 - #15
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	2	10% (10)	6 and 14	LO #1- #6 and LO #8 - #14
Summative assessment	Midterm Exam	2hr/2	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

	Material Covered
Week 1	Introduction to computer system
Week 2	Computer Basics: Identifying Computer Types, Understanding Software Types
Week 3	Computer Basics: Computer System Components, How Computers Communicate
Week 4	Parts of computer: Selecting and Using Input Devices, Selecting and Using Output Devices
Week 5	Parts of computer: Choosing Appropriate Storage Devices, Storage Problems
Week 6	Parts of computer : Processer
Week 7	Mid-term Exam
Week 8	Computer maintenance: format, system installation
Week 9	Operating System Basics: Understanding System Software, Comparing the Major Operating Systems
Week 10	Introduction to Windows 10: A Tour of Windows 10, Starting Up and Shutting Down
Week 11	Introduction to Windows 10: Running Applications, Managing Files
Week 12	Windows 10 Administration and Maintenance: Managing User Accounts, Using System Utilities, and Troubleshooting Common Windows Problems
Week 13	Understanding Application Software: Business Productivity Software, Graphics Software, Other Types of Software, and Managing Your Applications
Week 14	Understanding Microsoft Office 2010: Understanding the Office 2010 Interface, and Using Basic Features of Office Applications
Week 15	Understanding Microsoft Office 2010: Saving, Opening, and Creating Files, Printing and Sharing Files, Setting Application Options
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Lab 1: Describe the computer, data, information
Week 2	Lab 2: Display the computer types, Application software and system software
Week 3	Lab 3: Display the component of computer system: Hardware and Software.
Week 4	Lab 4: Display Input and Output devices
Week 5	Lab 5: Display the Internal Memory (ROM, RAM) and Secondary storage.
Week 6	Lab 7: Display the processor types
Week 7	Lab 7: Exam

Week 8	Lab 8: Applying the process of formatting the computer and installing the system
Week 9	Lab 9: Display software system: DOS and Windows and comparing between them
Week 10	Lab 10: Display Desktop elements, Start menu, Taskbar, Starting Up and Shutting Down.
Week 11	Lab 11 : Applied creating, opening, renaming, arranging, moving, copying, and deleting files.
Week 12	Lab 12 : Display Control Panel homepage, system backup, restore, desktop background, screen saver, sound, window color.
Week 13	Lab 13 : Setting time, date, language, region, mouse, internet and adding printer.
Week 14	Lab 14 : Applied Start MS 2010, Title bar, Ribbon, Status bar, Quick access toolbar
Week 15	Lab 15 : Applied saving, opening, creating files, printing and sharing files and setting application options

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	Computing fundamental: IC3 Edition, Copyright Year: 2014, John Wiley and Sons Ltd	Yes
Recommended Texts	step-by-step: Microsoft Office 2010, Copyright Year: 2011 Joyce Cox & Joan Lambert	No
Websites	Introduction to Computers and Office Productivity Software Course (HKUST) Coursera	

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.				

Discrete Structures				
معلومات المادة الدراسية				
Module Title	Discrete Structures		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☒ Seminar	
Module Code	MU010601103			
ECTS Credits	7			
SWL (hr/sem)	175			
Module Level	1	Semester of Delivery		1
Administering Department	Type Dept. Code	College	Type College Code	
Module Leader	Hazim Michman Trao		e-mail	hazimmichman@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification	Ph.D.	
Module Tutor			e-mail	
Peer Reviewer Name	Basim Akudaer Abbas	e-mail	baasim_math@uomustansiriyah.edu.iq	
Scientific Committee Approval Date		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. This course is an introductory text on a number of topics in discrete mathematics, intended primarily for students undertaking a first degree in computer science. 2. The material of this course has been divided into seven chapters covered on 15 weeks. 3. The basic areas covered: Sets, mathematical induction, Relations, Functions, Logic, Groups and Graphs. 4. Each chapter begins with a clear statement of pertinent definitions, principles and theorems together with illustrative.

	5. This is followed by sets of solved problems. The solved problems serve to illustrate and amplify the theory.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand sets and perform operations and algebra on sets. 2. Prove mathematical theorems using mathematical induction. 3. Determine properties of relations, identify equivalence and partial order relations, sketch relations. 4. Identify functions and determine their properties. 5. Define graphs, digraphs and trees, and identify their main properties. 6. Evaluate combinations and permutations on sets. 7. Analyze logical propositions via truth tables.
Indicative Contents المحتويات الإرشادية	The basic areas covered: Sets, mathematical induction, Relations, Functions, Logic, Groups and Graphs. • Introduce concepts of sets, and use the properties of set operations algebraically. • Use mathematical induction to prove some types of problem. • Investigate functions as relations and their properties. • Introduce mathematical logic for analyzing propositions and proving theorems. • Introduce basic concepts of graphs, digraphs and trees

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Assessment Strategy This course is designed to further the student's understanding of mathematics, to lead students to connect the mathematics to its application in computer science and to encourage the students to communicate their ideas and their expertise to the professional community. With this in mind, the course grade involves an assessment of their performance on in-class quizzes and exams that focus on the applications of discrete mathematics to computer science. • Weekly assigned homework to motivate students to do the work and earn credit accordingly. • Weekly in-class quizzes • Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation. • Three in-class exams to assess the student's accumulative mastery of content covered prior to time of exam. • A comprehensive final exam to assess the student's accumulative mastery of course material.

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	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	82	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	175		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	20% (20)	5 and 10	LO #1 - #2 and LO #4 - #5
	Assignments	2	10% (10)	2 and 12	LO #3 , #4 and #6
	Projects / Lab.				
	Report	1	10% (10)	13	All
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #4
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Set theory/ concept of a set, representation of sets, sub set, equality of sets, universal set
Week 2	Set theory/ empty set, disjoint sets, venn diagrams, set operations, union, intersection, Cartesian product, complements, differences, symmetric differences,
Week 3	Mathematical induction/ principle of mathematical induction, some examples of mathematical induction
Week 4	Graph theory/ sample graph, psedo graph, multi graph, sub graph, finite graph. propositions, biconditional propositions
Week 5	Graph theory/ degree of vertex, special families of graphs, complete graph, complete bipartite.
Week 6	Relations/ definition of relation, range of relation, domain of relation, representation of relations

Week 7	Relations/ inverse relation, number of relations, types of relations, reflexive, symmetric, anti-symmetric, transitive, and equivalence.
Week 8	Functions/ definition of function, range of function, domain of function
Week 9	Functions/ injection functions, surjections, composite functions, inverse function
Week 10	Mutual Inductance, Linear and Ideal Transformers, Circuits with Mutual Inductance
Week 11	Logic
Week 12	Logic
Week 13	Matrices/ definition of matrices, Square matrix, diagonal matrix, identity matrix, sum of matrices, multiplication of matrices, transpose of matrix
Week 14	Matrices/ some types of matrices, upper and lower triangle matrix, symmetric matrix, conjugate matrix, Hermitian matrix, skew Hermitian matrix, trace,
Week 15	Matrices/ Gaussian Elimination , Gaussian-Jordan, Inverse of matrices, Determinants of matrix,
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	
Week 2	
Week 3	
Week 4	
Week 5	
Week 6	
Week 7	

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Discrete Mathematics for New Technology /Second Edition/Rowan Garnier and John Taylor/ university of Brighton, UK 2002	Yes
Recommended Texts	Mathematics for Computer Science, Eerc Lehman and Tom Leighton 2004.	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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Computer Ethics		Module Delivery
Module Type	C		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	MU010601104		
ECTS Credits	4		
SWL (hr/sem)	100		
Module Level	1	Semester of Delivery	
Administering Department	Dept. of Computer Science	College	College of Science
Module Leader	Dr. Alaa A. Jabbar	e-mail	Sadco99@gmail.com
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification	Ph.D.
Module Tutor	Name (if available)	e-mail	E-mail
Peer Reviewer Name	Dr. Farah Qais	e-mail	Farah2qais@gmail.com
Scientific Committee Approval Date		Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	Semester	1
Co-requisites module	None	Semester	1

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	At the end of this course, students are expected to learn: 1. Concepts and tools that help them to acquire a set of skills and tools for evaluating ethical issues in this case will be able to identify and define the components of a structured plan for solving ethical problems and will be able to understand the basis for their own ethical system. 2. Guide and help the students to formulate a personal perspective towards ethical behavior socially. 3. Tools and methodology to develop the ethical analysis skills and professional values so they are able to decide by themselves if an action related to computer science is good or bad right or wrong. 4. The student will be enable to focus on the humanitarian , social, and cultural impact of computer technology by focusing on ethical issues faced by computing professionals , including those related to networking and the internet, intellectual property, privacy, security, computer risks and responsibilities
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. To Know about the Computer ethics "Morality + Ethical Decision Making in computing & Law + Purpose of Law + Morality and the Law". 2. To Know about the Computer Crimes " Types of Computer System Attacks", Computer Crime Prevention Strategies". 3. To Know about the "ACM Association of Computing Machinery", The ten commandments of Computer Ethics. 4. Explain of the Commandments of computer ethics, physical health issues 5. Accountability, Responsibility, and Liability 6. Explain of the "Foundations of Intellectual Property, Copyrights, Application for copyright, Duration of a copyright". 7. To Know about the Security, Ethics & Privacy 8. To Know about the Software piracy, protection of software under copyright Laws". 9. To Know about the Software Issues: Risks& Liabilities "Definitions & causes of software failures". 10. To Know about the Ethical, Privacy & Security Issues in the Online Social Network Ecosystems 11. To Know about the Elastic Extension Beyond the Traditional Computer, Mobile Devices Attacks 12. To Know about the Privacy and civil liberties, Legal protection of personal privacy

Indicative Contents المحتويات الإرشادية	<u>Part A Introduction to Computer Ethics</u> Computer ethics "Morality + Ethical Decision Making in computing & Law + Purpose of Law + Morality and the Law". Computer Crimes " Types of Computer System Attacks", Computer Crime Prevention Strategies". "ACM Association of Computing Machinery", The ten commandments of Computer Ethics. <u>Part B Commandments of computer ethics</u> Commandments of computer ethics, physical health issues Accountability, Responsibility, and Liability "Foundations of Intellectual Property, Copyrights, Application for copyright, Duration of a copyright". Security, Ethics & Privacy <u>Part C Security Issues</u> Software piracy, protection of software under copyright Laws". Software Issues: Risks& Liabilities "Definitions & causes of software failures". Ethical, Privacy & Security Issues in the Online Social Network Ecosystems Elastic Extension Beyond the Traditional Computer, Mobile Devices Attacks Privacy and civil liberties, Legal protection of personal privacy
---	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The field of ethics (or moral philosophy) involves systematizing, defending, and recommending concepts of right and wrong behavior. The philosophers today usually divide ethical theories into three general subject areas: met ethics, normative ethics, and applied ethics. Ethics deals with the standards of human conduct that direct the behavior of individuals and groups. These standards are developed by the society within which individual or groups exist. Every society has its rule about whether certain acts are accepted or not, and are often written into laws. Ethics are moral standards that help to guide behavior, actions, and choices. Actions must be guided by some ethical values to ensure that it maintain certain moral, social and legal standards.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
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	5	10% (10)	5 and 10	LO #1- #2 and #3- #4 and #5- #6 and #7-8 and #9- #11
	Assignments	2	5% (5)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	5% (5)	10	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	70% (70)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Computer ethics "Morality + Ethical Decision Making in computing & Law + Purpose of Law + Morality and the Law".
Week 2	Computer Ethics, New Win in an Old Bottle, Changing Premises, Different Temptations, Different Means of Delivery, Complacent Society, Ethical Muddles, Technology and Values.
Week 3	Computer Crimes " Types of Computer System Attacks", Computer Crime Prevention Strategies".
Week 4	"ACM Association of Computing Machinery", The ten commandments of Computer Ethics.
Week 5	Commandments of computer ethics, physical health issues.
Week 6	Accountability, Responsibility, and Liability.
Week 7	First exam
Week 8	"Foundations of Intellectual Property, Copyrights, Application for copyright, Duration of a copyright".
Week 9	Security, Ethics & Privacy.
Week 10	Software piracy, protection of software under copyright Laws".
Week 11	Software Issues: Risks& Liabilities "Definitions & causes of software failures".
Week 12	Ethical, Privacy & Security Issues in the Online Social Network Ecosystems.
Week 13	Elastic Extension Beyond the Traditional Computer, Mobile Devices Attacks.
Week 14	Privacy and civil liberties, Legal protection of personal privacy.
Week 15	Preparatory week before the final Exam
Week 16	Final Exam

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

Learning and Teaching Resources		
مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	1. Joseph Migga Kizza Ethics in Computing Undergraduate Topics in Computer Science, Department of Computer Science. University of Tennessee, Chattanooga, Tennessee, USA, ISBN 978-3-319-29104-8 ISBN 978-3-319-29106-2 (eBook), DOI 10.1007/978-3-319-29106-2, 2016. Publisher: Springer, 2016. 2. Ethics: A Global perspective by Dr.GiannisStamatellos ISBN-13:978-0763740849.1st edition Published :Jones&Bartlett learning , 2007. 3. Baasa, Sara (2012):A Gift of Fire: Social, Legal, and Ethical Issues for computing and technology(4thedition), Publisher: pearson, 2012, ISBN 10:0132492679.	Yes
Recommended Texts	The handbook of information and computer ethics, edited by: Kenneth EinarHimma and Herman T.Tavani, copyright@2008 by john wiley&sons, Inc.ISBN 978-0-471-79959-7(cloth).	available as PDF
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.				

Description for Ethics in Computer Science

Computer ethics is a field of applied ethics that addresses ethical issues in the use, design and management of information technology and in the formulation of ethical policies for its regulation in society

As young people grow to adulthood, the ethics they learn whilst in our care, will help secure our collective digital futures, reduce the likelihood of them taking a path into cybercrime, and use their values to make their decisions

Ethics is a worthwhile subject of study for at least three reasons: (1) studying ethics can help you appreciate ethical problems, (2) behaving ethically may help the long-term success of a business, and (3) behaving ethically is often correlated with behaving legally.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Democracy and human rights		Module Delivery	
Module Type	B		☒ Theory ☐ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU02			
ECTS Credits	2			
SWL (hr/sem)	50			
Module Level	1	Semester of Delivery		1
Administering Department	0003	College	0001	
Module Leader	M. M. Hussein Hizam Badr		e-mail	hussein_hezam@uomustansiriyah.edu.iq
Module Leader's Acad. Title		Module Leader's Qualification		
Module Tutor	Name (if available)		e-mail	E-mail
Peer Reviewer Name	Name		e-mail	E-mail
Scientific Committee Approval Date	01/06/2023		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- تعريف الطالب بما يعنيه مصطلح الديمقراطية والنظام الديمقراطي، فضلاً عن زيادة معرفة الطالب بالجانب المفاهيمي النظري والتطور التاريخي للديمقراطية. وكذلك تعريف الطالب بحقوق الإنسان وحرياته الأساسية، فضلاً عن زيادة معرفة الطالب بالجانب المفاهيمي النظري والتطور التاريخي لحقوق الإنسان. 2- تمكين الطالب من ممارسة حقوقه بصورة فعالة من خلال التنمية الشاملة لشخصيته بما يتفق وقيم المجتمع الديمقراطي وحقوق الإنسان . 3- تنمية مهارات وإدراك الطالب فيما يتعلق بالديمقراطية و حقوق الإنسان وتأثيراتها على المجتمع. 4- فهم واستنباط أبرز التجارب العالمية فيما يتعلق بالديمقراطية، ومحاولة استنباط ما يمكن تطبيقه في مجتمعاتنا المحلية. 5- التعرف بالدستور وبالانتخابات حقوقهم وواجباتهم وإبرز التجارب الانتخابية، وتحفيز الطلبة على أهمية المشاركة الفاعلة في الحياة السياسية
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1- التعرف بالديمقراطية وحقوق الإنسان 2- التعرف على أشكال وصور الديمقراطية وحقوق الإنسان 3- مراجعة جذور وتاريخ الديمقراطية وحقوق الإنسان عربياً وعالمياً 4- تسليط الضوء على الدستور والديمقراطية 5- معرفة ما تعنيه جرائم الإبادة الجماعية وماهي حقوق الضحايا وأهاليهم 6- التعرف على الضمانات التي تكفل النظام الديمقراطي
Indicative Contents المحتويات الإرشادية	يمثل التعليم والتنقيف في مجال الديمقراطية حقوق الإنسان عملية تدوم مدى الحياة وتنمي المعارف والمهارات كما انها تعزز الشخصية ووجه السلوك سعياً لتحقيق حماية مستدامة لحقوق الإنسان وحرياته، فضلاً عن ان التنقيف بحقوق الإنسان تسهم في صون كرامة البشر سعياً لإقامة مجتمعات تحترم وتقدر حقوق الإنسان

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	الاستراتيجية الرئيسة التي سيتم تبنيها في تقديم هذه المادة ستركز على تشجيع الطلبة على تقبل المواضيع المرتبطة بحقوق الإنسان، ومحاولة تحسين بعض المهارات المرتبطة بها، وسيتم تحقيق ذلك من خلال البرامج التعليمية التفاعلية او من خلال اتباع بعض الطرق تدريس الحديثة.
-------------------	--

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعاً

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	33	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعياً	2
--	----	--	---

Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	17	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	3
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	50		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	70% (70)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	مفهوم ومعنى الديمقراطية مفهوم ومعنى حقوق الانسان
Week 2	اشكال وصور الديمقراطية وحقوق الانسان
Week 3	الخلفية التاريخية للديمقراطية وحقوق الانسان
Week 4	سمات النظام الديمقراطي ومزاياه وعلاقته بحقوق الانسان
Week 5	المكونات الرئيسية للديمقراطية
Week 6	الدستور حقوق الانسان والديمقراطية
Week 7	Mid-term Exam + Unit-Step Forcing, Forced Response, the RLC Circuit
Week 8	حقوق الانسان العراقي - الدستور الدائم لجمهورية العراق 2005
Week 9	الانتخابات والسلوكيات التي تضمن حق الناخب والمنتخب
Week 10	مؤسسات المجتمع المدني وحقوق الانسان والديمقراطية
Week 11	العلاقة بين حقوق الإنسان والديمقراطية
Week 12	جرائم الابادة الجماعية وبعض النماذج العالمية
Week 13	ضمانات الحريات والحقوق العامة
Week 14	الضمانات التي تكفل النظام الديمقراطي

Week 15	الحكم الرشيد
Week 16	Preparatory week before the final Exam

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	محاضرات في حقوق الانسان والديمقراطية	نعم
Recommended Texts	حقوق الانسان والديمقراطية والحريات العامة	نعم
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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Arabic Language		Module Delivery	
Module Type	Basic		☒ Theory ☐ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU06			
ECTS Credits	2			
SWL (hr/sem)	50			
Module Level	1	Semester of Delivery		1
Administering Department	05	College	01	
Module Leader	Dr. Entiha Abbas Aliwi		e-mail	Enteaaabbas@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assist. Professor		Module Leader's Qualification	Ph.D.
Module Tutor	No available		e-mail	E-mail
Peer Reviewer Name			e-mail	E-mail
Scientific Committee Approval Date	01/11/2023		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- مساعدة الطالب على حب اللغة العربية لغة القرآن الكريم. 2- التعرف على مواطن الجمال في اللغة العربية وآدابها. 3- أن يكتسب الطالب القدرة على دراسة فروع اللغة العربية. 4- تعريف الطالب بألفاظ اللغة العربية الصحيحة وتراكيبها وأساليبها السليمة بطريقة مشوقة وجذابة. 5- تنمية الطالب مهارات اللغة السليمة في المخاطبات الرسمية
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1- تعلم خصائص اللغة ومميزاتها عن اللغات الأخرى 2- تحقيق الكفاءات اللغوية والقواعدية في أداء الطلبة للغة العربية في كل من الانشاء والقواعد والاملاء 3- تشخيص مواطن الاعراب في مادة اللغة العربية 4- معرفة الحركات (الفتحة والضمة والكسرة) واصوات أخرى مثل همزة الفصل والوصل. 5- قراءة الكلمات والجمل وفهم معانيها والربط بينها 6- اكتساب الطالب القدرة على التعبير الصحيح في التخاطب، والتحدث، والكتابة. 7- تعلم اساسيات كتابة الاعداد 8- يتضح لديه علامات الاصلية والفرعية للجمل العربية 9- رفع مستوى الطالب على القراءة الصحيحة، والنطق السليم، وفهم الأفكار 10- تعليم الطالب بحفظ قدر كافٍ من النصوص الأدبية، وأن يستمتع بها، ويتذوق مظاهر الجمال فيها
Indicative Contents المحتويات الإرشادية	جلسة الإرشادية تحقيق فرصة المسترشد ليعبر عما بداخله ويسلك بحرية فعلى المرشد حسن الاصغاء ودقة الملاحظة وتركيز الانتباه لكل قول أو فعل كي يدرك المرشد الجوانب المختلفة للمشكلة والمحافظة على اللغة الفصحى والابتعاد عن العامية ومعالجة الأخطاء الشائعة.

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	التعليم هو انتقال الطلاب من مرحلة التركيز على المهارات بالدرجات الابتدائية إلى مرحلة التركيز على تبني وأن الطلاب يقوموا بمواجهه كثير من المطالب من أجل قراءة المعلومات عبر الكتب الدراسية، ويقوموا كذلك بتدوين الملاحظات أثناء المحاضرات، كما يتم العمل بطريقة مستقلة، بجانب التعبير.
-------------------	---

Student Workload (SWL)

الحمل الدراسي للطلاب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	35	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	15	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	1
Total SWL (h/sem) الحمل الدراسي الكلي للطلاب خلال الفصل	50		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	4 and 14	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	5 and 9	LO #2, #3 and #7, #8
	Projects / Lab.	1	10% (10)	12	LO #5, #6 and #9
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	نبذة مختصرة عن أهمية اللغة
Week 2	محاضرة عن اقسام اللغة العربية
Week 3	محاضرة عن اقسام الجملة العربية في لغتنا
Week 4	محاضرة عن اقسام الجملة الاسمية والفعلية في اللغة
Week 5	محاضرة عن كتابة العدد
Week 6	محاضرة عن جمل التميز
Week 7	محاضرة كيفية وضع علامات التنقيط
Week 8	<u>Midterm Exam</u>
Week 9	محاضرة عن الادب العربي
Week 10	محاضرة عن اقسام الادب العربي
Week 11	محاضرة عن اقسام الشعر العربي مع نموذج من قصائده
Week 12	محاضرة عن اقسام النثر مع نموذج من نثره
Week 13	محاضرة تضم تطبيقات عملية لكل المواد اللغة
Week 14	(محاضرة) اختبارات عملية لكل مواد الادب
Week 15	معرفة الهمزة المتوسطة والمتطرفة
Week 16	Preparatory week before the final Exam

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	محاضرات في اللغة العربية العامة	نعم
Recommended Texts	اللغة العربية لغير الاختصاص	نعم
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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Programming Fundamentals (II)		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	MU010601205		
ECTS Credits	7		
SWL (hr/sem)	175		
Module Level	1	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader	Nada Thanoon Ahmed	e-mail	nada_thanoon@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant. Prof	Module Leader's Qualification	MSc.
Module Tutor	Zied Othman Ahmed	e-mail	zied_othman@uomustansiriyah.edu.iq
Peer Reviewer Name	Dr. Jamal Nasir Hasoon	e-mail	jamal.hasoon@uomustansiriyah.edu.iq
Scientific Committee Approval Date	01/06/2023	Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Programming Fundamentals (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 focuses on the increased levels of care and attention in the teaching of structured programming. This course will be covering: Arrays. string and string processing. Methods and parameter passing. Structured decomposition. File input and output methods, and Analyze and explain the behaviors of simple programs involving the fundamental programming constructs covered by this unit using C# programming language. Furthermore, modify and expand short programs that use standard conditional and iterative control structures and functions.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand the Method, Array, strings and file in C# language. 2. Understand the Object –Oriented paradigms. 3. Be able to write a program in C# language using array statement. 4. Be able to write a program using method statement and introduction to object-oriented paradigms. 5. Be able to write a program using file input and output. 6. Apply computer programming concepts to problems or situations, and demonstrate appropriate design, coding, testing, and documenting of computer programs that solve these problems or situations using concept of structured programming.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. 1. Arrays and Lists: [30 hrs] ○ Introduction to arrays and lists. ○ Array/List declaration and initialization. ○ Array/List manipulation (accessing elements, adding, removing). ○ Array/List traversal and searching. ○ Multidimensional arrays. 2. Methods and Modular Programming: [34 hrs] ○ Introduction to Methods. ○ Method declaration and definition. ○ Method parameters and return values. ○ Scope and lifetime of variables. ○ Modular programming concepts. 3. Strings and Characters: [12 hrs] ○ String manipulation and operations. ○ String comparison and searching.

	○ String concatenation and splitting. ○ Character manipulation. 4. Input and Output: [12 hrs] ○ Standard input and output. ○ Streams ○ File input and output. ○ Reading from a file ○ Writing to a File ○ Formatting output. ○ Error handling and exception handling.
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning Strategies: 1. Hands-on Practice: Programming is a practical skill that requires practice. Allocate dedicated time to work on coding exercises, small projects, and problem-solving activities. 2. Code Reading and Analysis: Study and analyze existing code examples and programs written by experienced developers. Understand the logic, structure, and flow of the code to enhance your understanding of programming concepts. 3. Break Problems into Smaller Parts: When faced with complex programming problems, break them down into smaller, manageable tasks. This approach, known as "divide and conquer," helps tackle the problem systematically. 4. Collaborative Learning: Engage in discussions and collaborations with fellow learners or join programming communities and forums to share knowledge, ask questions, and solve problems collectively. 5. Debugging and Troubleshooting: Debugging is an essential skill in programming. Practice identifying and fixing errors in your code, as it enhances your problem-solving abilities and helps you understand the logic behind the code. Teaching Strategies: 1. Conceptual Explanation: Start by providing clear and concise explanations of programming concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2. Hands-on Coding Exercises: Include practical coding exercises and assignments that reinforce the concepts taught. Encourage students to apply the theory into practice, helping them develop their coding skills.

	- Visualizations and Diagrams: Use visual aids like diagrams, flowcharts, or pseudocode to illustrate programming concepts and algorithms. Visual representations can aid understanding, especially for beginners. - Interactive Coding Sessions: Conduct live coding sessions where students can actively participate, ask questions, and solve problems alongside the instructor. This approach promotes engagement and practical learning. - Resources and References: Share relevant programming resources, recommended textbooks, and online tutorials to supplement the curriculum and encourage further exploration.
--	---

Student Workload (SWL)			
الحمل الدراسي للطلاب محسوب ل ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	94	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	81	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	5
Total SWL (h/sem) الحمل الدراسي الكلي للطلاب خلال الفصل	175		

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, #3, #4, #6
	Assignments	2	8% (8)	4 and 12	LO #1, #3,#4,#6
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3, #4, #5.#6
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4, #6
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Array: One dimension array
Week 2	One dimension array
Week 3	Two-dimension array
Week 4	Two-dimension array
Week 5	Two-dimension array
Week 6	Methods
Week 7	Methods
Week 8	Midterm Exam / Method overloading
Week 9	Method overloading
Week 10	Working with structures
Week 11	Working with structures
Week 12	Strings and Characters
Week 13	Strings and Characters
Week 14	Input and Output: file input and output
Week 15	Input and Output: file input and output
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Examples One dimension array
Week 2	Examples 1D Array
Week 3	Examples Two-dimension array
Week 4	Examples Two-dimension array
Week 5	Examples Two-dimension array
Week 6	: first Exam / Examples Methods
Week 7	Examples Methods
Week 8	Examples Methods
Week 9	Examples Methods
Week 10	: second exam : Examples Structures.
Week 11	Examples Structures.

Week 12	Examples Strings and Characters
Week 13	Examples Strings and Characters
Week 14	Examples Input and Output: file input and output
Week 15	Examples Input and Output: file input and output

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	- "Programming Basics with C#" Book and Video Lessons Svetlin Nakov and TeameBook 2019 - Visual C# How to Program, 6th edition Published by Pearson (July 14th 2021) - Copyright © 2017	Yes
Recommended Texts	- Deitle and Deitle. "C Sharp How to Program", Prentic Hall, 2009	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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Computer Skills II		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010601206			
ECTS Credits	7			
SWL (hr/sem)	175			
Module Level	1	Semester of Delivery		2
Administering Department	Computer science	College	College of science	
Module Leader	Salah Taha		e-mail	Salah.taha@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Asst. Professor		Module Leader's Qualification	MSc.
Module Tutor			e-mail	
Peer Reviewer Name	Yasamin Hamza		e-mail	yhamza@uomustansiriyah.edu.iq
Scientific Committee Approval Date	01/06/2023		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	The course is intended to teach the students the basics and principles of Microsoft Office 2010 by learning and understanding the basic Features of Office Applications such as: 1. Word. 2. PowerPoint. 3. Excel. 4. Access.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	The student is expected to learn the following at the end of this course. 1. Understanding the Word 2010 Interface 2. Learn how to write and arrange reports 3. Learn how to insert the tables and images in reports 4. Learn how to type with all fingers 5. Understanding and using PowerPoint. 6. Learn how to prepare and arrange slides. 7. Learn how to insert and arrange tables and pictures on slides 8. Learn how to display slides and their contents. 9. Understanding the Excel 2010 interface. 10. Learn how to set up and format tables. 11. How to insert and use functions. 12. Understanding the Access 2010 interface. 13. Learn how to design interfaces. 14. Learn how to create tables and link them
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. Part A - Word 2010 Word 2010 Interface, write and arrange reports, insert the tables and images in reports, and write with all fingers. [25 hrs] Part B - Excel 2010 prepare and arrange slides, insert and arrange tables and pictures on slides, and display slides and their contents. [25 hrs] Part C - PowerPoint Excel 2010 interface, create and format the tables, and insert and use functions. [22 hrs] Part D - Access. Understanding the Access 2010 interface, designing interfaces, creating tables, and linking them. [22 hrs]

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	In this course, students will be urged to participate in the lectures by preparing reports on each topic and presenting them during the lecture in order to increase the information related to the lecture. In addition, the students ask to complete projects related to the course material and a competition will be held among students to choose the best projects.
-------------------	---

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	94	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	6
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	81	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	5
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	175		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	3	10% (10)	4,8 and 12	LO #1- #3 and LO #4- #6 and LO #9 - #11
	Assignments	2	10% (10)	5 and 10	LO #1- #4 and LO #5 - #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	2	10% (10)	8 and 14	LO #1- #7 and LO #8 - #14
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #6
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Microsoft Word 2010 (Editing and Formatting Text, and Formatting Paragraphs)
Week 2	Formatting Documents and Sections, and Working with Pictures
Week 3	Working with Tables, and Correcting and Collaborating
Week 4	Microsoft PowerPoint 2010 (Explore PowerPoint 2010, Work with Slides, and Work with Slide Text)
Week 5	Format Slides, and Review and Deliver Presentations
Week 6	Insert Tables, and Animation and Sound and Movies
Week 7	Mid-term Exam + Microsoft Word 2010 ,and Microsoft PowerPoint 2010
Week 8	Microsoft Excel 2010 (Understanding the Excel Interface, Data and Excel Tables, and Calculations on Data)
Week 9	Shortcut key in Excel2010, Specific Data by Using Filters, Reordering and Summarizing Data
Week 10	Data and Multiple Sources, and Creating Charts and Graphics
Week 11	Using the functions
Week 12	Microsoft Access 2010 (Explore an Access 2010 Database, Databases and Simple Tables, and Simple Forms and display data)
Week 13	Query and selected query, Query Data, and Report
Week 14	Create Relationships, and One-to-many Relationships
Week 15	Create Relationships, Many – to-many Relationships, Simple Reports
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Applied editing and Formatting Text and Formatting Paragraphs
Week 2	Lab 2: Applied formatting Documents and Sections, add picture, insert table
Week 3	Lab 3: Applied writing with all fingers
Week 4	Lab 4: Applied work with Slides
Week 5	Lab 5: Applied format Slides
Week 6	Lab 7: Applied add Tables, and add animation
Week 7	Lab 7: Exam

Week 8	Lab 8: Applied data and Excel Tables, performing Calculations on Data , and shortcut key
Week 9	Lab 9: Applied focusing on Specific Data by using filters, combining Data from Multiple Sources
Week 10	Lab 10: Applied adding charts and graphics
Week 11	Lab 11 : Apply the functions
Week 12	Lab 12 : Applied create databases and Simple Tables, Create Simple Forms, Create Simple Reports
Week 13	Lab 13 : Applied create data base selected query, and report
Week 14	Lab 14 : Applied create a data base using all relationships (One-to-one Relationships, and One-to-many Relationships)
Week 15	Lab 15 : Applied create (Many – to-many Relationships), and create report

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Computing fundamental: IC3 Edition, Copyright Year: 2014, John Wiley and Sons Ltd	Yes
Recommended Texts	step-by-step: Microsoft Office 2010, Copyright Year: 2011 Joyce Cox & Joan Lambert	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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Logic Design		Module Delivery	
Module Type	Core		☒ Theory ☐ Lecture ☐ Lab ☒ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010601207			
ECTS Credits	6			
SWL (hr/sem)	150			
Module Level	UGI	Semester of Delivery		2
Administering Department	Computer Science	College	College of Science	
Module Leader	Dr. Bassam Basim Jamil		e-mail	dr.balkindy@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant Professor		Module Leader's Qualification	Ph.D.
Module Tutor	Name (if available)		e-mail	E-mail
Peer Reviewer Name	Dr. Basim Jamil Ali		e-mail	basimja6nd@uomustansiriyah.edu.iq
Scientific Committee Approval Date		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. To understand the basic and relative numbering systems (e.g., decimal, binary, octal, and hexadecimal, and other systems). 2. To focus on designing logical circuits built from using the logical gates using Boolean algebra 3. To understand simplified logical circuits built from using the logical gates using Boolean algebra and K-Map will open the gate to simplify logical circuits. 4. To understand the design of combinational circuits such as decoder, encoder, multiplexer, counters, etc. 5. To develop problem solving skills and understanding of circuit theory through the application of techniques. 6. To understand the sequential logic circuits such as flip-flops.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Knowledge of different number systems (Decimal, Binary, Octal, and Hexadecimal) 2. Understand the conversions between numbering systems. 3. Knowledge of the basic logic gates. 4. Understand how to simplify boolean expressions using the rules of Boolean algebra. 5. Knowledge on the express of Sum of Minterms and Product of Maxterms. 6. Understand the structure of Karnaugh Map, for simplification, of different map size up to 6 inputs, and don't care condition. 7. Knowledges on how to design different combinational logic circuits. 8. Knowledges on sequential logic circuits.
Indicative Contents المحتويات الإرشادية	Part A – Numbering Systems, Complement, and Arithmetic Operations [20 hrs] Numbering systems [15 hrs] – System's base or radix – Decimal system and numbers – Counting on numbering systems – Binary system – Octal system – Hexadecimal system – Binary codes: gray codes, BCD code, 84-2-1, Excess+3 – ASCII character codes – Conversion from decimal system to other systems – Conversion from other systems to decimal system – Conversions from binary codes from/to other systems Arithmetic Operations [5 hrs] – Addition on systems – Direct subtraction – Subtraction using complement – Multiplication, and division.

	Part B – Binary Logic, Logical gates, and Canonical Forms [10 hrs] – Design of logical gates: OR, AND, NOT, XOR, XNOR, NAND, and NOR gates – Design of logic gates with truth tables – Simplification using Boolean Algebra – Complement functions ($r-1$ and r complements) – Boolean function evaluation – Canonical and standard forms: Standard forms using (OR) and (AND) gates only – Simplifications using K-Map of 2, 3, 4, 5, or 6 inputs – Don't care condition – Simplification of Sum of Products and product of sums using K-Map
	Part C – Design of Combinational Logic Circuits [30 hrs] – Design procedure – Half and full adder – Half and full subtractor – Code conversion – Binary parallel adder – BCD adder – Decoder – Encoder – Multiplexer – Read only memory (ROM)
	Part D – Sequential Logic Circuits [15 hrs] – RS Flip Flops – D Flip-Flops – T and JK Flip-Flops – Registers

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The primary approach to delivering the logic design module will focus on fostering active student participation and enhancing their critical thinking abilities. This will be accomplished through a combination of interactive lectures, engaging lab sessions, and the inclusion of various simple experiments that involve sampling activities, specifically designed to captivate student interest. The aim is to encourage students to actively engage with the material, refine their problem-solving skills, and expand their overall understanding of logic design concepts. By implementing these strategies, the learning experience will be enriched and students will be better equipped to grasp the subject matter effectively.

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب ل ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	71	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	150		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	3	8% (8)	2, 6, and 9	LO #1-#2, #3-#6, and #7
	Assignments	2	6% (6)	4 and 12	LO #1-#3, and #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	8% (8)	13	LO #3-#7
Summative assessment	Midterm Exam	2hr	8% (8)	7	LO #1 - #6
	Final Exam/ Lab	1 hr	10% (10)	16	All
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to numbering systems: Decimal system, Binary system, Octal system and Hexadecimal system.
Week 2	Introduction to number base conversion: Conversion from decimal system to another system, Conversion from another system to decimal, conversion between all number system.
Week 3	Arithmetic Operations: binary arithmetic operations and arithmetic operation in other systems. The Complement: r and (r-1) complement, Subtraction with (r) and (r-1) complement.
Week 4	Binary codes: BCD, Excess-3, Gray code and other codes. Binary Logic and Logic Gates: Basic gates and Boolean algebra. Boolean Function evaluation and complement Functions.
Week 5	Implement the Boolean Function: Some of Minterms (S.O.M & Product of Maxterms (P.O.M). Minimization of logic circuits using K-Map using 2, 3, 4 and 5 variables in Product of Sum Simplification and Sum of Product Simplifications forms.
Week 6	Don't care conditions: Simplification the Boolean functions in S.O.P and P.O.S forms using don't care

	conditions. Basic Gates with NAND and NOR gates. Implement Boolean Function using NAND and NOR gates.
Week 7	Mid-term Exam
Week 8	Introduction to combinational logic circuits. Design the basic combinational logic circuits: Adders: design half and full adder, Binary Parallel Adder, BCD adder
Week 9	Combinational logic circuits: design the basic combinational logic circuits: Subtractors: Design half and full subtractor circuits using r and (r-1) complement.
Week 10	Combinational logic circuits: design the basic combinational logic circuits: Decoder and Encoder
Week 11	Combinational logic circuits: design the basic combinational logic circuits: Multiplexer and Demultiplexer.
Week 12	Combinational logic circuits: design different logic combinational circuits: Code converter, Read Only Memory (ROM), and programmable Logic Array (PLA)
Week 13	Sequential Logic Circuits: Flip Flops (RS, D Flip-Flops)
Week 14	Sequential Logic Circuits: Flip Flops (T and JK Flip-Flops).
Week 15	Sequential Logic Circuits: Registers.
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Lab 1: Basic logic Gates (AND, OR, and NOT gates)
Week 2	Lab 2: logic Gates (NAND, NOR)
Week 3	Lab 3: combination gates with examples
Week 4	Lab 4: Digital logic circuits analysis and converting Boolean expressions to digital circuits
Week 5	Lab 5: Boolean algebra and Simplification of Boolean expressions (Part I)
Week 6	Lab 6: Boolean algebra and Simplification of Boolean expressions (Part II)
Week 7	Lab 7: Mid-Term Exam
Week 8	Lab 8: DE Morgan's Theory
Week 9	Lab 9: DE Morgan's Theory and the Universal Gates (part I)
Week 10	Lab 10: DE Morgan's Theory and the Universal Gates (part II)
Week 11	Lab 11: XOR and XNOR gates: Basics and Applications
Week 12	Lab 12: Implementing Half-Adder and Half-Subtractors circuits
Week 13	Lab 13: Implementing Full-Adder and Full-Subtractors circuits
Week 14	Lab 14: Simplification of Boolean expressions – II
Week 15	Lab 15: Review on Logic circuits
Week 16	Course Final Exam

Learning and Teaching Resources		
مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	M. M. Mano, "Digital Design", Prentice Hall, 2010.	No
Recommended Texts	A.F.Kana, "Digital Logic Design", 2012	No
Websites	Online video lectures: https://www.youtube.com/watch?v=N9NNmjpZaGg&list=PLICTN_xRLDd7k06SMWNpx_FLrqkk1fcVt	

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.				

Module Information				
معلومات المادة الدراسية				
Module Title	Mathematics		Module Delivery	
Module Type	C		☒ Theory ☒ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☒ Seminar	
Module Code	MU010601208			
ECTS Credits	5			
SWL (hr/sem)	125			
Module Level	1	Semester of Delivery		2
Administering Department		College		
Module Leader	Basim Akudaer Abbas		e-mail	baasim_math@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Lecturer		Module Leader's Qualification	Ms
Module Tutor	Name (if available)		e-mail	
Peer Reviewer Name	Hazim Michman Trao		e-mail	hazimmichman@uomustansiriyah.edu.iq
Scientific Committee Approval Date		Version Number	1.0	

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	Semester	1
Co-requisites module	None	Semester	1

Module Aims, Learning Outcomes and Indicative Contents	
أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Demonstrate knowledge of basic precalculus concepts and skills. 2. Work with functions and limits. 3. Recognize continuity and use the properties of continuous function. 4. Solve problems using derivatives. 5. Find derivatives of algebraic and trigonometric functions. 6. Definition and rules of differentiation to differentiate.

	- Sketch the graph of function using asymptotes. - Evaluate integrals both by using Riemann sums and by using the Fundamental Theorem of Calculus. - Evaluate integrals using advanced techniques of integration, such as inverse substitution, partial fractions and integration by parts.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	- Concepts of basic functions - Interpret equations and graphs of the basic classes of functions - Identify trigonometric functions and their features - Analyze inverse functions - Examine exponential, logarithmic, and hyperbolic functions - Determine the existence of estimate numerically and graphically and find algebraically the limits of functions - Recognize and determine infinite limits and limits at infinity and interpret with respect to asymptotic behavior. - Determine continuity at a point or on intervals and distinguish between the types of discontinuities at a point. Definite and Indefinite Integration - Determine the derivative of a function using the limit definition. Interpret the derivative as the slope of a tangent line to a graph, the slope of a graph at a point, and the rate of change of a dependent variable with respect to an independent variable - Determine the derivative and higher derivatives of a function explicitly using differentiation formulas. - Determine derivatives implicitly. - Solve related rates problems. - Determine absolute extrema for a continuous function on a closed interval. Use these and other appropriate techniques to solve optimization problems. - Use the first and second derivatives to analyze and sketch the graph of a function, including asymptotes, intervals on which the graph is increasing, decreasing, concave up, or concave down, and any local extrema and inflection points. - Determine antiderivatives and indefinite integrals and integrate by substitution. - Use the Fundamental Theorem of Calculus to evaluate definite integrals. - Use definite integrals to find areas of planar regions.
Indicative Contents المحتويات الإرشادية	The basic areas covered: Functions, Continuity, Derivative, and Integrals. - Introduce concepts of Functions, and use the properties of functions as value of function, domain of a function, Graph of a function and operations with Functions. - Learn some techniques for Computing Limits, Infinite Limits, Limits at Infinity - Learn how to evaluate the Continuity of function. - Investigate some techniques for Computing Derivatives of trigonometric Functions. - Introduce definition of definite Integral, techniques of integration

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	Assessment Strategy This course is designed to further the student's understanding of mathematics, to lead students to connect the mathematics to its application in computer science and to encourage the students to communicate their ideas and their expertise to the professional community. With this in mind, the course grade involves an assessment of their performance on in-class quizzes and exams that focus on the applications of discrete mathematics to computer science. • Weekly assigned homework to motivate students to do the work and earn credit accordingly. • Weekly in-class quizzes • Weekly, in-class presentations by students of solutions to real world problems related to the course material and classroom discussion and critique of the presentation. • Three in-class exams to assess the student's accumulative mastery of content covered prior to time of exam. • A comprehensive final exam to assess the student's accumulative mastery of course material.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

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	62	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
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	20% (20)	5 and 10	LO #1- #5 and #10- #14
	Assignments	2	10% (10)	2 and 12	LO #6, #7 and #8, #9
	Projects / Lab.				
	Report	1	10% (10)	13	All
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #8
	Final Exam	3hr	50% (50)	16	All

Total assessment	100% (100 Marks)		100% (100 Marks)
------------------	------------------	--	------------------

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Identify Functions; Find the Value of a Function; Find the Domain of a Function; Use operations with Functions
Week 2	Identify the Graph of a Function and use Information about the Graph
Week 3	Identify and use Properties of Functions
Week 4	Graph the Functions in the Library of Functions; Graph Piecewise-defined Functions
Week 5	Quiz
Week 6	Identify the Graph of trigonometric function,
Week 7	Techniques for Computing Limits, Infinite Limits, Limits at Infinity
Week 8	Continuity
Week 9	Identify derivative, techniques of differentiation
Week 10	quiz
Week 11	Techniques for Computing Derivatives of trigonometric Functions
Week 12	Theorems, examples to recite on demand all indeterminate forms of a limit (0/0, infinity/infinity, mixed product, indeterminate differences, indeterminate powers
Week 12	Derivatives of Logarithmic
Week 13	Exponential Function
Week 14	Interpretations and Properties of indefinite Integrals
Week 15	definition of definite Integral, techniques of integration
Week 16	Final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	
Week 2	
Week 3	
Week 4	
Week 5	

Week 6	
Week 7	

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	1. Calculus: late transcendental. - 7 th Edition combined /Howard Anton, Irl Bivens. Stephen Davis. 2. Exercises and Problems in Calculus ,John M. Erdman Portland State University Version August 1, 2013 c 2010 John M. Erdman	Yes
Recommended Texts	Calculus Volume 1 SENIOR CONTRIBUTING AUTHORS EDWIN "JED" HERMAN, UNIVERSITY OF WISCONSIN-STEVENS POINTGILBERT STRANG, MASSACHUSETTS INSTITUTE OF TECHNOLOGY	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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Information Technology		Module Delivery	
Module Type	C		☒ Theory ☒ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010601209			
ECTS Credits	4			
SWL (hr/sem)	100			
Module Level	1	Semester of Delivery		2
Administering Department		College		
Module Leader	Dr. Farah Qais		e-mail	Farah1qais@gmail.com
Module Leader's Acad. Title	Lecturer		Module Leader's Qualification	Ph.D.
Module Tutor	Name (if available)		e-mail	E-mail
Peer Reviewer Name	Dr. Alaa A. Jabbar		e-mail	Sadco99@gmail.com
Scientific Committee Approval Date		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. learn the Components of a computer system 2. Learn the HARDWARE that includes (CPU, Control Unit, Arithmetic Logic Unit (ALU), Registers, Network devices, Memory primary, secondary storage, and Memory Capacity

	- Learn the software that includes the Operating system and Applications Software (Word processing, Spreadsheets, Databases Presentation, Accounts / Payroll, Web browsing, Web authoring, and a multimedia application. - The course also gives the details for the Types of Networks (LAN, (MAN) Metropolitan area Networks, and WAN. Also learn the Categories of Networks such as (Peer-to-Peer Networks and Server-Based Networks) - Learn the uses of computer applications in business, within government, hospitals (the health care system), and in education. - Acquires a variety of IT-related information that will be useful to him/ / her for studying in the computer field
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	- Understand the general term of Information technology(IT): and to know the difference between Information System and Data, Information , Knowledge. - Understand the knowledge about the Computer–Based Information System and the Computer Hierarchy and also The description of (Supercomputers , Mainframe Computers , Minicomputers , Networked computers , Portable computer, laptop, notebook, palmtop and PDA personal digital assistant.). - Understand the knowledge and the descriptions of (Personal Digital Assistant (PDA) and PC) - understand the Graphical User Interface with her advantages and the - Components of a computer system - understand the complete description for the HARDWARE which includes - Processing devices(CPU) ,Control Unit, Arithmetic Logic Unit (ALU), and Registers - Network devices, Memory with the capacity, primary and secondary storage and the Input output devices - know the difference between internal and external hard disks - The basic parameters that determine the performance of a given hard disk drive - The brief description about the back up your computer and format your computer - know the complete description for the SOFTWARE which include Operating system and Applications Software (Word processing ,Spreadsheets ,Databases, Presentation, Accounts / Payroll, Web browsing ,Web authoring multimedia application , and Desktop publishing) know about the Human Being and the systems development? understand the Main Parts of a Personal Computer - know The system unit (case): and The System (Mother) Board Chipset (Northbridge and Southbridge) - know and understand the difference between Memory (Ram) and Memory (Rom) and know - The ROM-BIOS and flash BIOS -know The types of Monitors ❖ Flat screen monitors ➤ VDU(visual display unit) ✓ CRT stands for (cathode ray tube).(available raster scan and random vector) ➤ LCD stands for (liquid crystal display). know the Computer performance and the factors affecting overall computer speed

	16. learn the Programming language(Machine Language, Second-generation languages(Assembly) and also know the High-level language which includes (Procedural language, Nonprocedural languages and Natural languages (Visual programming languages and object-oriented programming languages)) 17. know the Types of program translator and know the differences between them • Assembler with Advantages and disadvantages • Compiler with Advantages and disadvantages • Interpreter with Advantages and disadvantages 18. know the description of Computer Network and the Types of Networks: (LAN, MAN Metropolitan Area Network and WAN) 19. Give the description of Categories of Network (Peer-to-Peer Networks and Server-Based Networks) 20. Advantages and Disadvantages of workgroup computing 21. The knowledge about the Internet, Intranet , Extranet and the World Wide Web (WWW) 22. Learn the teleworking with the Advantages and disadvantages
Indicative Contents المحتويات الإرشادية	<u>Part A general term of information technology</u> - the general term Information technology(IT): and the Information System: Data, Information , Knowledge. - the Computer-Based Information System and the Computer Hierarchy with the description of Supercomputers, Mainframe Computers, Minicomputers, Networked computers , Portable computer, laptop, notebook, palmtop, and PDA personal digital assistant. - the descriptions of Personal Digital Assistant (PDA),, the PC, the Graphical User Interface with her advantages and the -Components of a computer system <u>Part B HARDWARE components</u> A- complete HARDWARE which includes - Processing devices(CPU), Control Unit: Arithmetic Logic Unit (ALU) Registers. - Network devices ,Memory, primary and secondary storage with • Network devices • Memory with capacity • primary and secondary storage • Input and output devices • The internal and external hard disks • The basic parameters determine the performance of a given hard disk drive • back up and format your computer <u>Part C SOFTWARE components</u> • Operating system, - O.S, Applications Software which includes (Word processing, Spreadsheets, Databases, Presentation, Accounts / Payroll, Web browsing , Web authoring, multimedia application AND Desktop publishing • THE High-level language • Procedural language • Nonprocedural languages • Natural languages(Visual programming languages and object-oriented programming languages) • Types of program translator

	- Assembler with Advantages and disadvantages - Compiler with Advantages and disadvantages - Interpreter with Advantages and disadvantages Part D Information Networks AND Computers at Work 1- Computer Network and Types of Networks: • LAN • (MAN) Metropolitan area Network • WAN 2- Categories of Network (Peer-to-Peer Networks and Server-Based Networks) 3- The workgroup with Advantages and Disadvantages 4- The Internet, Intranet and Extranet 5- The World Wide Web (WWW) 6- The tele-working with the Advantages and disadvantages 7- Computers at Work includes Uses of the computer applications in • Business • Hospitals and the health care system • Education • Uses of large scale computers applications within government
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	During the 11-13 weeks of the course, The learning and Teaching Strategies will cover the general idea of Information Technology (IT) that relates to the use of computers to create and maintain data. Also, learn the role of Information technology related to all aspects of managing and processing information, especially within a large organization.

Student Workload (SWL) الحمل الدراسي للطلاب محسوب ل ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطلاب خلال الفصل	48	Structured SWL (h/w) الحمل الدراسي المنتظم للطلاب أسبوعيا	3
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطلاب خلال الفصل	52	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطلاب أسبوعيا	3
Total SWL (h/sem) الحمل الدراسي الكلي للطلاب خلال الفصل	100		

Module Evaluation تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	5	10% (10)	5 and 10	LO #1- #2 and #3- #4 and #5- #6 and #7-8 and #9- #11
	Assignments	2	5% (5)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	5% (5)	10	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	70% (70)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Chapter 1 Understand the terms hardware, software, Information Technology (IT) 1- Information technology(IT): 2- Information System Data, Information , Knowledge 3- Computer –Based Information System
Week 2	4- Computer Hierarchy - Super computers - Mainframe Computers - Minicomputers - Networked computers - Portable computer, laptop, notebook, palmtop, PDA personal digital assistant. 5- Personal Digital Assistant (PDA) 6- The PC 7- Graphical User Interface with her advantages
Week 3	Chapter Two HARDWARE AND SOFTWARE <i>Part one</i> • -Components of a computer system A- HARDWARE • Processing devices(CPU) , Control Uni, Arithmetic Logic Unit (ALU) performs arithmetic calculations and Registers • Network devices
Week 4	• Memory with the capacity • primary and secondary storage • Input and output devices ❖ The internal and external hard disks ❖ The basic parameters determine the performance of a given hard disk drive ❖ back up and format your computer

Week 5	B - SOFTWARE a - Operating system - The O.S with the famous operating systems b- Applications Software Word processing , Spreadsheets ,Databases , Presentation , Accounts / Payroll , Web browsing , Web authoring , multimedia application and Desktop publishing. - The Human Being - The systems development
Week 6	Chapter two Main Parts of a Personal Computer Part 2 Main Parts of a Personal Computer 1- The system unit (case): 2- The System (Mother) Board • Chipset (Northbridge and Southbridge)
Week 7	First exam
Week 8	3- Memory (Ram) 4- Memory (Rom) - The ROM-BIOS - flash BIOS 5- Hard (Fixed) Disk
Week 9	6- The Monitor Flat screen monitors ➤ VDU(visual display unit) ➤ CRT stands for (cathode ray tube). ✓ available raster scan ✓ random vector ➤ LCD stands for (liquid crystal display)
Week 10	Chapter Three Computer performance Computer performance 1- Factors affecting overall computer speed 2- Programming language - Machine Language - Second –generation languages(Assembly)
Week 11	3- High-level language • Procedural language • Nonprocedural languages • Natural languages(Visual programming languages and object-oriented programming languages) 4- Types of program translator - Assembler with Advantages and disadvantages - Compiler with Advantages and disadvantages - Interpreter with Advantages and disadvantages
Week 12	Chapter Four Information Networks 8- Computer Network and Types of Networks: • LAN • (MAN) Metropolitan area Network

	• WAN
	9- Categories of Network (Peer-to-Peer Networks and Server-Based Networks)
Week 13	10- The workgroup with Advantages and Disadvantages 11- The Internet, Intranet and Extranet 12- The World Wide Web (WWW) 13- The tele-working with the Advantages and disadvantages
Week 14	Chapter five Computers at Work Uses of the computer applications in • Business • Hospitals and the health care system • Education • Uses of large scale computers applications within government
Week 15	Preparatory week before the final Exam
Week 16	Final Exam

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

Learning and Teaching Resources مصادر التعلم والتدريس		
		Available in the Library?
Required Texts	1. <i>"Basic Concepts of Information and Communication Technology, notes"</i>, Dario Ilija Rendulić , 2011. 2. <i>"Computational Science: Ensuring America's Competitiveness"</i>, Marc R. Benioff Edw and Lazowska , 2005. 3. <i>Basic Concepts Of Information Technology (IT)</i>, Dheeraj Mehrotra, 4. <i>Introduction to Information Technology—Hardware, Software, and Telecommunications</i>, Information Technology for the Health	Yes(available as PDF)

	Professions, Third Edition, by Lillian Burke and Barbara Weill. Published by Prentice Hall. Copyright © 2009 5. <i>Collection of papers in IT from the internet.</i>	
Recommended Texts		No
Websites	https://www.comptia.org/content/guides/what-is-a-wide-area-network https://www.vedantu.com/computer-science/uses-of-computers-in-hospitals https://www.wheelhouseit.com/computer-application-in-healthcare/	

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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	English Language		Module Delivery	
Module Type	Basic		☒ Theory ☐ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU01			
ECTS Credits	2			
SWL (hr/sem)	50			
Module Level	1	Semester of Delivery		2
Administering Department	05	College	01	
Module Leader	Farah A. Raheem		e-mail	E-mail
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification	M A English Literature	
Module Tutor	No available		e-mail	E-mail
Peer Reviewer Name	Dr. Ali Raheem	e-mail	aliraheem@uomustansiriyah.edu.iq	
Scientific Committee Approval Date	01/06/2023	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. Urging students to acquire and revise their previous knowledge of the basics of English language in general. 2. Urging students to learn and master the use of English grammar for the first seven units of the curriculum approved by the Iraqi Ministry of Higher Education. 3. Enabling students to acquire, improve and practice the various skills of English language (listening, speaking, reading and writing) for the first seven units of the curriculum approved by the Iraqi Ministry of Higher Education. 4. Enabling students to use English language in their academic study of science instead of the Arabic language to achieve scientific progress. 5. Enabling students to communicate in English with others in their daily life inside/outside the campus.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1- Students will be able to speak English as fluently as possible. 2- Students will be able to listen and understand English conversations, talks or lectures in English better than before. 3- Students will be able to read and comprehend written works such as scientific books, lectures, reports and papers in English easier than before. 4- Students will be able to write in English better than before, such as lab reports, explanations, and definitions . 5- Students will be able to use English grammar correctly more than before. 6- Students will be able to use their improved English language skills in their academic career more than ever before
Indicative Contents المحتويات الإرشادية	<u>Unite 1 . Hello (4 hr.)</u> Grammar :V. to be , This is... Vocabulary: Greetings , Numbers, Introduction. Everyday English: Greetings & conversation <u>Unite 2 . Your world (4 hr.)</u> Grammar : Subject pronouns, V. to be in questions Vocabulary: Countries, Nouns, Adjectives Everyday English: Numbers, Countries. <u>Unite 3 . All about you (4 hr.)</u> Grammar : V. to be in questions, Negative, & Short Answers Vocabulary: Jobs & personal Information& interviews.

	Everyday English: Social Expressions <u>Unite 4 . Family and friends (4 hr.)</u> Grammar : Possessive adjectives, Possessive 's , has/have& Adjective + noun Vocabulary: The family & Describing a friend Everyday English: The alphabet , On the phone, & Saying email addresses <u>Unite 5 . The way I live (4 hr.)</u> Grammar : Present Simple, a and an & Adjective + noun. Vocabulary: Sports/ Food/ Drinks, Adjectives & Languages and nationalities. Everyday English: How much is it? , Numbers1-100 & prices. <u>Unite 6 . Every day (4 hr.)</u> Grammar : Present Simple, Questions and negatives & Adverbs of frequency. Vocabulary: The time, Verbs and nouns & Words that go together. Everyday English: Days of the week & Prepositions of time <u>Unite 7 . My favourites (4 hr.)</u> Grammar : Question words ,Pronouns(Subject/Object/Possessive) & this and that Vocabulary: Adjectives, Opposite adjectives, & Places Everyday English: Can (questions)
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	During the 14- 15 weeks of the first semester, The following learning and Teaching Strategies will be used : 1- Beginner students need more careful staged approach with plenty of drillings, repetition, and revisions. So, a step by step technique will be effective. 2- Use question technique to stimulate discussion, emphasize the value of students' answers, and encourage active participation. 3- Emphasizing and improving previously acquired English language skills and abilities. 4- Encouraging learners to work together in chat groups to improve their speaking skills. Ask them to do a written homework to improve their writing skills as well.

	5- Make use of <i>iTools</i> or <i>New Headway Plus Beginner Digital Edition</i>, videos and audio tracks using the smart board to motivate learners to hear and acquire better English pronunciation especially to improve listening and speaking skills. 6- Using social media, YouTube channels and reliable applications to facilitate the independent learning process and make it more fun. 7- Encourage learners to participate in language activities, online courses and international language tests such as TOEFL and IELTS as an optional extracurricular activity. 8- Linking different English language skills with learners' other scientific curriculums by presenting short scientific lectures in English on smart boards and asking them to answer questions, write reports, prepare oral presentations as homework, or participate in a group discussion inside the class
--	---

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب لـ 15 اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	33	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	17	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	1
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	50		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative	Quizzes	1	10% (10)	10,12	LO #3, #4 and # 5

assessment	Assignments	1	10% (10)	14	LO #4, #5, and #6
	Listening Exam (Project)	1	10% (10)	2,4,10,12	LO #1, and #2
	Oral Discussions (Report)	1	10% (10)	All	All
Summative assessment	Midterm Exam	2hr	10% (10)	1-6	All
	Final Exam	3hr	50% (50)	All	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introductory lecture: "English Language: Importance, Concepts, & Methods of Teaching". Notes about <i>New Headway plus : beginner</i> Unit -1- Student's book : Hello! WHAT'S YOUR NAME? p.6 INTRODUCTIONS p.7 HOW ARE YOU? p.8 EVERYDAY ENGLISH: Good morning! p.9 VOCABULARY AND SPEAKING: What's this in English? p.10-11
Week 2	Unit -1- Workbook: Hello! INTRODUCTIONS p. 4 Nice to meet you P.5 Vocabulary: p.6-7

	EVERYDAY ENGLISH: Good morning! p.8 Do not forget! p.9 Writing : p.10-11
Week 3	Unit -2- Student's book : Your world SHE'S FROM JAPAN p.12 QUESTIONS p.13 PRACTICE : Cities and countries p.14-15 READING AND SPEAKING : Where are they from? p.16 EVERYDAY ENGLISH : Numbers 11-30 p.17
Week 4	Unit -2- Workbook: Your world Countries p.12 Am/are/ is p.13 Listening: At a wedding p. 14 Questions p. 14 Adjectives p. 14 Reading : Where are they p.15 Listening p. 16 Everyday English : Numbers 11-30 p. 16 Do not Forget! p.17 Writing : p.18-19
Week 5	Unit -3- Student's book : All about you Jobs p. 18 Negatives and questions p.18

	Personal information p.19 STUDENTS WALK FOR CHARITY p.20 Practice p. 21 READING AND LISTENING : The charity walk p.22 EVERYDAY ENGLISH : Social expressions (1 p.23
Week 6	Unit -3- Workbook: All about you Jobs p.20 Questions and negatives : is/isn't p.21 Negatives and questions : I'm not/they aren 't p.22 Listening p.23 Vocabulary p.23 Everyday English : Social expressions (1) p. 24 Do not forget p.25 Writing: p.26-27
Week 7	Midterm Exam
Week 8	Unit -4- Student's book : Family and Friends MY FAMILY : Possessive' S p.24-25 Practice p. 26 Annie's brother p.27 Practice p. 28 READING AND WRITING : My best friend p28-29 EVERYDAY ENGLISH : The alphabet p.30 On the phone p.31

Week 9	Unit -4- Workbook: Family and friends Possessives p.28-29 Vocabulary p.29 Alfie and his family p.30 Reading : Who's happy? p.31 Vocabulary p.32 Everyday English :The alphabet p.32 Do not forget p.33 Writing: p.34-35
Week 10	Unit -5- Student's book : The Way I Live Sports/Food/Drinks p. 32 Present Simple -I/you/we/they p.32 -33 Reading and speaking : COLIN BRODIE FROM DUNDEE p.34-35 VOCABULARY AND PRONUNCIATION : Languages and nationalities p.36-37 LISTENING AND SPEAKING : At a dinner Party p.38 Everyday English p.39
Week 11	Unit -5- Workbook: The Way I Live Sports, food, drink p.36 Things I like p.37 Present Simple : The swimmer Michael Phelps p.38 LISTENING p.39 Everyday English : How much is it? p.40

	Do not forget p.41 Writing: p. 42-43
Week 12	Unit -6- Student's book : Every day The time p.40 WHAT TIME DO YOU GET UP? p.41 Elliot's DAY p.42-43 PRACTICE : Lois's day p. 44-45 VOCABULARY AND SPEAKING : Words that go together p.46 EVERYDAY ENGLISH : Days of the week p.47
Week 13	Unit -6- Workbook: Every day The time p.44 Present Simple p.45-46 Pronunciation p.47 Prepositions p.47 Vocabulary p.47 Reading: Barack Obama, America's first black President p.48 EVERYDAY ENGLISH : Days of the week p.49 Writing: p. 50-51
Week 14	Unit -7- Student's book : My favourites Question words p.48-49 THIS IS MY FAVOURITE ... p.50

	PRACTICE : I like them! p.51 VOCABULARY : Adjectives p. 52 READING AND WRITING : A postcard from San Francisco p.53 EVERYDAY ENGLISH p.54-55
Week 15	Unit -7- Workbook: My favourites Questions p.52-53 Pronouns and possessives p.54 this/ that p.54-55 Vocabulary : Adjectives : happy/miserable p.55 EVERYDAY ENGLISH p.56 Don't forget! p.57 Writing: p. 58-59
Week 16	Final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
	Non

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	1- Soars, John and Liz. <u>New Headway Plus : Beginner Student's Book</u>, 1st Edition, Oxford University Press, Oxford , (2014). Digital Edition , With DVD- ROM. Required Units 1-7 only. Soars, John and Liz. <u>New Headway Plus : Beginner Workbook without Key</u>, 1st Edition, Oxford University Press, Oxford , (2014). Digital Edition , With new writing Micro- skills. Required Units 1-7 only.	Both Books are available as printed copies & PDF Notes : Most of PDF and printed copies are different from the required edition. Besides, some printed editions and PDF of the Workbook are with Key.
Recommended Texts	1- Murphy, Raymond. <u>English Grammar In Use</u>. 3rd Edition, Oxford University Press, Oxford,(2012). 2- Soars, John and Liz & Merifield, Sue. <u>Headway Beginner Teachers' Guide. 5th Edition</u>, Oxford University Press, Oxford , (2019). (For Teachers)	Available as a printed copy & PDF Available as a PDF
Websites	1- Oxford University Press/English Language Teaching Worldwide, https://elt.oup.com/, an online website supports teachers who teach Headway plus Beginner text books. 2- BBC Learning English , https://www.bbc.co.uk /learning English , an online website also available as an application on Google play, and social media sites. It presents free different programs to improve English Language Skills.	

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	FX – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
(0 – 49)	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.

Program Structure Annex No. (3)

Lab	Theoretical		Module Code	Module Name	Level One
2	3	هياكل البيانات	MU010602101	Data Structures	Semester 1
2	3	البرمجة الكيانية	MU010602102	Object Oriented Programming	
	3	النظرية الاحتمالية	MU010602103	Theory of Computation	
2	3	تنظيم ومعمارية الحاسوب	MU010602104	Computer Organization and Architecture	
2	3	تصميم المواقع	MU010602105	Web Design	
	3	هندسة البرامجيات	MU010602106	Software Engineering	
	2	اللغة الانجليزية (2)	MU02	English Language (II)	

Lab	Theoretical		Module Code	Module Name	Level One
2	3	تصميم وتحليل الخوارزميات	MU010602201	Algorithms Design and Analysis	Semester 2
2	3	قواعد البيانات	MU010602202	Database Fundamentals	
2	3	مترجمات	MU010602203	Compiler	
2	3	البرمجة بلغة بايثون	MU010602204	Python Programming	
	2	الإحصاء والاحتمالية	MU010602205	Statistics and Probability	
	2	اللغة العربية	MU010602206	Arabic Language	
	2	جرائم حزب البعث	MU02	The Crime of Baath Regime in Iraq	

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Data Structures		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010602101			
ECTS Credits	5			
SWL (hr/sem)	125			
Module Level	UGII	Semester of Delivery		3
Administering Department	Computer Science	College	College of Science	
Module Leader	Dr. Bassam Basim Jamil		e-mail	Dr.balkindy@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant. Prof		Module Leader's Qualification	
Module Tutor	Assistant.Prof Bassam Basim Jamil		e-mail	Dr.balkindy@uomustansiriyah.edu.iq
Peer Reviewer Name			e-mail	
Scientific Committee Approval Date	/ /2024		Version Number	1.0

Relation with other Modules				
العلاقة مع المواد الدراسية الأخرى				
Prerequisite module	Programming Fundamentals (I)		Semester	1
	Programming Fundamentals (II)		Semester	2
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 : 1- strategies and techniques that are required to store data (Data Structures) and perform operations on such data. 2- To teach efficient storage mechanisms of data for an easy access. 3- To design and implementation of various basic and advanced data structures. 4- To introduce various techniques for representation of the data in the real world. 5 - To develop application using data structures. Topics covered include this course : Survey basic abstract data types, their associated algorithms, and how they are implemented. structures of stacks, queues, lists, linked list and tree ; performance tradeoffs of different implementations; and asymptotic analysis of running time and memory usage.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completion of the course, the students will be able to: 1- Student will be able to choose appropriate data structure as applied to specified problem Definition. 2- Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures. 3- Students will be able to use linear and non-linear data structures like stacks, queues, linked list , tree . 4- Compare and contrast the costs and benefits of dynamic and static data structure implementations. 5- Students will be able to Convert the types of Data structures to programs develop their coding skill.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I : [8 hrs] Introduction to Data Structures , How to choose the suitable data structure Types of data structures , Memory Representation ,Introduction to Abstract Data Type , Introduction to Time Complexity (Space Complexity, Time Complexity) UNIT-II : [22 hrs] Stack ADT definition and operation , applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation. Queue ADT definition and operations, Circular queues Insertion and deletion operations. UNIT-III : [28 hrs]

	- Array list ADT definition, Array list operations. - Linked List ADT definition , Linked List operations. -Linked List Design Modification : - Circular Linked List , Circular Linked List Operations, Traversing Circular Linked List. - Doubly Linked List . Doubly Linked List Operations , Traversing Doubly Linked List . - Linked Stack, Linked Queue, Linked Circular Queue Operations. - Recursion UNIT IIII : [15 hrs] Trees – Definitions, tree representation, properties of trees, Binary tree, Types of Binary tree Binary tree representation, binary tree properties, binary tree traversals, binary tree implementation, applications of trees. Binary Search Tree: Building a Binary Search tree , Finding Minimum and Maximum values in Binary Search tree , Delete value in Binary Search Tree Representation of Arithmetic Expressions using Binary Tree
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<u>Learning Strategies:</u> The best way to learn and remember Data Structures is to practice them. 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 Data structures exercises , small projects, and problem-solving activities. 2- Reading types of data structures : Study and analyze existing data structures examples and programs written by experienced developers. write the programs enhance your understanding of Data structures . 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. <u>Teaching Strategies:</u>

	1- Conceptual Explanation: Start by providing clear and concise explanations of Data Structures concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2- Hands-on Coding Exercises: Include practical Data Structures exercises and assignments that reinforce the concepts taught. Convert the types of Data structures to programs . Encourage students to apply the algorithm into practice, helping them develop their coding skill. 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 Data Structures 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	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	46	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	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)	6 and 10	LO #1, #2, #3, #4, #5
	Assignments	2	8% (8)	4 and 12	LO #1, #2, #3, #4, #5
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3, #4, #5
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4, #5
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to Data Structures: - How to choose the suitable data structure - Types of data structures - Memory Representation - Introduction to Abstract Data Type
Week 2	- Introduction to Time Complexity : - Space Complexity - Time Complexity (Big –O Notation)
Week 3	Stack - The Stack Abstract Data Type - Array Stack - Stack Operations - Time Complexity of these operations
Week 4	Applications of the stack: - Check for balancing of parenthesis - Convert infix expression to postfix - Evaluate postfix expression - Use of stack in Function calls
Week 5	Queue - The Queue Abstract Data Type - Queue operations - Time Complexity of these operations
Week 6	Circular Queue and Priority Queues: - The Abstract Data Type - Operations - Time Complexity of these operations
Week 7	Midterm Exam / Lists : - Array list - The array List Abstract Data Type - Time Complexity of these operations - Array list operation
Week 8	Linked List - Storage Allocation - Pointers - Linked List Abstract Data Type

	- Traversing a Linked List
Week 9	Linked List: - Linked List Operations - Time Complexity of these operations
Week 10	Linked List Design Modification : - Circular Linked List - Circular Linked List Operations - Traversing Circular Linked List - Time Complexity of these operations
Week 11	Linked List Design Modification : - Doubly Linked List - Doubly Linked List Operations - Traversing Doubly Linked List - Time Complexity of these operations
Week 12	Linked Stack, Linked Queue, Linked Circular Queue - Operations - Time Complexity of this operations - Recursion
Week 13	Binary Tree: - Types of Binary Tree - Traversing Binary Tree: (post order, in order, preorder)
Week 14	Binary Search Tree: - Building a Binary Search tree - Finding Minimum and Maximum values in Binary Search tree
Week 15	Binary Search Tree: - Delete value in Binary Search Tree - Representation of Arithmetic Expressions using Binary Tree
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	Review of VC# language
Week 2	Programming of stack
Week 3	Examples of stack
Week 4	Programming of Queue with Quiz
Week 5	Programming of Circular Queue
Week 6	Programming of Circular Queue
Week 7	Midterm Exam / Programming of array list

Week 8	Programming of array list
Week 9	Programming of Linked List
Week 10	Programming of Linked List With quiz
Week 11	Examples on Linked List
Week 12	Programming of Recursion Factorial and power, Sum and Fibonacci
Week 13	Programming of Binary Search Tree
Week 14	Programming of Binary Search Tree
Week 15	Programming of Binary Search Tree

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	[1]: Author: MICHAEL McMillan. Title:" Data Structures and Algorithms Using C#" , 2007 [2]: Author : Thomas H. Cormen , CHARLES E. LEISERSON Title :“Introduction to Algorithms “, third edition ,2009	Yes
Recommended Texts	[1]: Author: Bruno R. Preiss Title:" Data Structures and Algorithms with Object-Oriented Design Patterns in C# " 2001 [2]: Author: Deitle and Deitle. Title: "C Sharp How To Program", Prentic Hall, 2009	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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Object-Oriented Programming		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010602102			
ECTS Credits	5			
SWL (hr/sem)	125			
Module Level	UGII	Semester of Delivery		3
Administering Department	Computer Science	College	College of Science	
Module Leader	Dr. Hassan Kassim		e-mail	hassan@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification		
Module Tutor	Dr. Mohammad Azeez		e-mail	Mohammada.abdulhassan@uomustansiriyah.edu.iq
Peer Reviewer Name	Dr. Hassan Kassim		e-mail	Mohammada.abdulhassan@uomustansiriyah.edu.iq
Scientific Committee Approval Date	12 /5 /2024		Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Programming Fundamentals (I)	Semester	1
	Programming Fundamentals (II)	Semester	2
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 : 1- strategies and techniques of OOP. 2- To teach principles of formal languages. 3- To design and implementation of various basic Automata and machines. 4- To introduce various techniques for mathematical models of languages. 5 - To develop different types of Automata. Topics covered include this course: This subject provides the concepts and techniques of object-oriented programming by using Microsoft Visual C#. These techniques help students to build simple object-oriented programs using visual C#.Net platform to build robust and flexible systems.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completion of the course, the students will be able to: 1.To increase students' capabilities in order to develop high levels and robust computer systems and principles of mathematical models of formal languages and Automata. 2. Student will be able to choose appropriate OOP as applied to specified problem Definition. 3. Student will be able to handle operations of OOP. 4. Students will be able to use FA, NFA, Moore, Mealy PDA, Turing Machines . 5. Students will be able to Convert the types of Automata to programs and develop their coding skill.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I : [8 hrs] Introduction: Introduction to O.O. paradigm, Genealogy of object oriented languages: structured programming, procedural programming, Advantages of OOP. Object Oriented Analysis (OOA): Defining abstract data type (ADT) Pillars of OOP: Abstraction, encapsulation, inheritance, and polymorphism Unified Modeling Language (UML) UNIT-II : [22 hrs] Concepts: Classes and Objects, Class members: Data members (fields) and member functions (methods), Class member visibility (private, public, protected), Class variables and instance variables, Class methods and in- stance methods, Formulate Class Diagram Constructors Single and multiple constructors with and without parameters Object initialization UNIT-III : [28 hrs] Methods: Method and operator overloading, Method overriding, Method visibility (public, private, protected), Abstract classes, Define relationships: associations, aggregation

	UNIT IIII : [15 hrs] Class Hierarchy Single inheritance, Polymorphism Multiple inheritance, Subclasses (derived classes) Super classes, (base classes) Creating and deleting objects (instances) Creating array of objects Invocation of superclass methods and constructors. Objects vs. variables Classes Objects vs. types Collection classes and Class libraries
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<u>Learning Strategies:</u> The best way to learn and remember OOP is to practice them. 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 OOP exercises , small projects, and problem-solving activities. 2- Reading types of OOP : Study and analyze existing OOP examples and programs written by experienced developers. write the programs enhance your understanding of OOP . 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. <u>Teaching Strategies:</u> 1- Conceptual Explanation: Start by providing clear and concise explanations of OOP concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2- .Hands-on Coding Exercises: Include practical OOP exercises and assignments that reinforce the concepts taught. Convert the types of OOP to programs . Encourage students to apply the algorithm into practice, helping them develop their coding skill. 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 OOP 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	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	45	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	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)	6 and 10	LO #1, #2, #3, #4, #5
	Assignments	2	8% (8)	4 and 12	LO #1, #2, #3, #4, #5
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3, #4, #5
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to OOP: - Introduction to O.O. paradigm, Genealogy of object oriented languages: structured programming,

	procedural programming, Advantages of OOP.
Week 2	- Object Oriented Analysis (OOA): Defining abstract data type (ADT) Pillars of OOP: Abstraction, encapsulation, inheritance, polymorphism Unified Modeling Language (UML)
Week 3	Concepts: Classes and Objects, Class members: Data members (fields) member functions (methods),
Week 4	Class member visibility (private, public, protected), Class variables and instance variables, Class methods and in- stance methods, Formulate Class Diagram
Week 5	Constructors Single and multiple constructors with and without parameters Object initialization
Week 6	Methods: Method and operator overloading, Method overriding,
Week 7	Midterm Exam
Week 8	Method visibility (public, private, protected), Abstract classes, Define relationships: associations, aggregation
Week 9	Class Hierarchy Single inheritance, Polymorphism
Week 10	Second Exam
Week 11	Multiple inheritance, Subclasses (derived classes) Super classes, (base classes)
Week 12	Creating and deleting objects (instances) Creating array of objects
Week 13	Invocation of superclass methods and constructors. Objects vs. variables Classes
Week 14	Objects vs. types Collection classes and Class libraries
Week 15	Polymorphism Static polymorphism Dynamic polymorphism

Delivery Plan (Weekly Lab. Syllabus) المناهج الاسبوعي للمختبر	
	Material Covered
Week 1	Visual C# review: Conditional and Looping statements Variables and data types.
Week 2	Visual C# review: Methods with and without parameters.
Week 3	Working with Classes Construct a class, Add/del/access data, members (get, set)
Week 4	Working with Classes Instance variables, Class methods, Access class methods, Array of class objects
Week 5	Work with Constructors single and multiple, constructor(s) with or without parameters
Week 6	Working with Methods Method Overriding and Overloading
Week 7	First exam
Week 8	Working with Methods working with multiple classes
Week 9	Working with single and multiple inherent classes.
Week 10	Second Exam
Week 11	Class Hierarchy Multiple inheritance
Week 12	Class Hierarchy Aggregation, association relations, super class command
Week 13	Creating and deleting objects, array of objects.
Week 14	Examples for Programs to produce OO program
Week 15	Examples for Programs to produce OO program

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	[1]: Troelsen, Andrew. Pro C# 5.0 and the .NET 4.5 Framework. Apress, 2012. [2]: Kolling, Michael, and David J. Barnes. Objects first with Java: A practical introduction using BlueJ. (2005).	Yes
Recommended Texts	[1]: Simon Kendal, <i>Object oriented programming using C#</i> , © 2011 Simon Kendal & bookboon.com, ISBN 978-87-7681-814-2	Yes
Websites	https://classroom.google.com/u/1/c/NjUxNDYwNzk1OTkw	

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.				

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	
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	<u>Learning Strategies:</u> 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.

	<u>Teaching Strategies:</u> 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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Computer Organization & Architecture		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☒ Seminar
Module Code	MU010602104		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	UGII	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader	Sundos Abdulameer Hameed	e-mail	ss.aa.cs@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant Prof Doctor	Module Leader's Qualification	PHD.
Module Tutor	Muna Abdulhussain Radi	e-mail	muna.ali@uomustansiriyah.edu.iq
Peer Reviewer Name		e-mail	
Scientific Committee Approval Date	20/04/2024	Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Logic Design (1)	Semester	2
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 will describe the basics of 1. Topics include computer system performance, instruction set architectures, pipelining, branch prediction, memory-hierarchy design, and a brief introduction to multiprocessor architecture issues. General Introduction of Emulator 8086 and assembly language in the lab. 2. Assembly language programming, including logic, comparing and branching, interrupts, macros, procedures, arrays, program design, testing, debugging, loading, and linking; combinational, arithmetic and logical circuits ALU; memory circuits, latches, flip-flops, registers; computer structure; fetch-execute cycles, clocks and timing; microprogramming and microarchitecture; data path, timing, sequencing; cache memory organization; RISC architectures
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand the key components within a computer system and organization. 2. Understand number systems and implement binary arithmetic with overflow handling and floating-point numbers. 3. understand digital logics, data path, and control using logic gates, combinational and sequential components 4. Analyze the cause of pipeline hazards and implement ways to overcome them. 5. Understand memory hierarchy and apply the principle of locality in cache design. 6. Understanding the need for the use of memory hierarchy (cache, main memory, storage devices) to ensure the design of a balanced computer system. 7. Understanding the fundamental principles of CPU and control unit design. 8. Discuss the relationship between the design of a computer system and the design of an operating system to operate it. 9. Evaluate various trade-offs in designing the instruction set architecture.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. 1. GENERAL INTRODUCTION & REVIEW: ○ Basic concepts revision ○ computer Architectures ○ computer design, ○ History of computers ○ computer Components 2. DATA REPRESENTATION ○ Data Type Representation Signed number, fixed, floating point, character ○ Addition Ripple Carry Adder, Carry Look-ahead,

	○ Multiplication Shift and Add, Booth Multiplier, Carry Save Multiplier 3. GENERAL PURPOSE REGISTERS ○ 8086 Emulator ○ Combinational and sequential circuit ○ Instruction set 4. MAIN MEMORY ORGANIZATION ○ Type of computer memory ○ Memory expansions ○ Addressing modes and effective address ○ Hierarchical memory organization ○ Direct Memory Access 5. CENTRAL PROCESSING ○ CPU Organization & Operation ○ Hardware Implementation ○ Performance of Computer ○ Basic Measurement Metrics ○ Pipelining: instruction 6. PARALLEL PROCESSORS ○ Parallel Program ○ SISD, MIMD, SIMD, SPMD, and Vector ○ HW Multithreading ○ Multicore and Shared Memory Multiprocessors
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Learning Strategies: 1. Understand Fundamentals: Start with the basics. Learn about binary arithmetic, logic gates, and basic digital circuit design. Understanding these fundamental concepts will provide a solid foundation 2. Follow a Structured Curriculum Look for structured courses or textbooks on computer architecture. These resources typically start with the basics and gradually introduce more complex topics. 3. Hands-On Practice: Theory is essential, but hands-on practice can deepen your understanding. Experiment with circuit simulators, assembly language programming, and hardware description languages. 4. Utilize Visual Aids: Computer architecture often involves abstract concepts that can be challenging to visualize. Look for diagrams, animations, or interactive simulations that illustrate key concepts like CPU pipelines, memory hierarchies, and instruction execution. Teaching Strategies:

	Start with Basics: Begin by laying a solid foundation with fundamental concepts such as binary arithmetic, logic gates, and Boolean algebra. Use real-world examples to illustrate abstract concepts and make them more relatable. Use Visual Aids: Computer architecture often involves intricate diagrams and structures. Utilize visual aids such as slides, animations, and interactive simulations to help students visualize abstract concepts like CPU pipelines, memory hierarchies, and instruction execution. Encourage Active Learning: Instead of simply lecturing, engage students in active learning activities such as group discussions, problem-solving exercises, and hands-on projects. Provide Real-World Examples: Relate computer architecture concepts to real-world applications and industry practices. Show how concepts like caching, pipelining, and parallel processing are used in modern processors and systems to improve performance and efficiency. Demonstrate with Simulations: Use software tools and simulators to demonstrate how different architectural designs and optimizations impact system performance. Interactive simulations allow students to explore complex concepts in a controlled environment and see cause-and-effect relationships in action. Integrate Practical Exercises: Design hands-on exercises and lab assignments that reinforce theoretical concepts and allow students to apply their knowledge in a practical setting.
--	--

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	46	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	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)	5 and 7	LO #1, #2, #3,#4
	Assignments	2	8% (8)	11 and 15	LO #5,#6, #7,#8
	Projects / Lab.	1	14% (14)	Continuous	All

	Report	1	8% (8)		LO #8, #9
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #4, #5, #6, #7
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	General Introduction & Review
Week 2	Data Representation
Week 3	Combinational and sequential circuit
Week 4	Main Memory Organization
Week 5	Hierarchical memory organization
Week 6	Memory expansions and Instruction formats
Week 7	Central Processing Organization
Week 8	Midterm Exam
Week 9	Execute Of Complete Instruction
Week 10	Performance of Computer
Week 11	Pipelining processing
Week 12	Direct Memory Access
Week 13	Input-Output Organization
Week 14	RISC and CISC pipeline, SISD, MIMD, SIMD, SPMD, and Vector
Week 15	Huffman code instruction
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	General Introduction of Emulator 8086
Week 2	Numbering System
Week 3	assembly language (part1)
Week 4	assembly language (part2)
Week 5	assembly language (part3)

Week 6	8086 interaction set1
Week 7	8086 interaction set2
Week 8	Examples Methods
Week 9	Examples Methods
Week 10	Variables
Week 11	Inside the CPU
Week 12	Memory Access
Week 13	XCHG Instruction
Week 14	Arithmetic Instruction
Week 15	Register Flag

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	- "Wang, Shuangbao Paul. Computer Architecture and Organization: Fundamentals and Architecture Security. Springer Nature, 2021. - Zomaya, Albert Y. "Advanced computer architecture and parallel processing." (2021)..	Yes
Recommended Texts	- Hayes, John P. <i>Computer architecture and organisation</i>. Tata McGraw Hill, 2022. - Nisan, Noam, and Shimon Schocken. <i>The elements of computing systems: building a modern computer from first principles</i>. MIT press, 2021. - Null, Linda. <i>Essentials of Computer Organization and Architecture</i>. Jones & Bartlett Learning, 2023.	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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Web Design and Programming		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010602105			
ECTS Credits	5			
SWL (hr/sem)	125			
Module Level	UGII	Semester of Delivery		3
Administering Department	Computer Science	College	College of Science	
Module Leader	Dr. Asmaa Sadiq		e-mail	asmaasadiq@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant Prof.	Module Leader's Qualification		
Module Tutor	Dr. Eman Hato		e-mail	emanhato@uomustansiriyah.edu.iq
Peer Reviewer Name	Dr. Asmaa Sadiq		e-mail	asmaasadiq@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	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	This subject presents the website design and programming that is essential topic to both Computer Science and Information Technology. This course focuses on fundamentals of design and coding of website using HTML. Scope of this course could summarize by: Editing and Viewing HTML Files, Setting Up the Document Structure, Formatting Text by Using Tags, Using Lists and Backgrounds, Creating Hyperlinks and Anchors, Style Sheets and Graphics, Page Layout and Navigation, Creating Tables, Formatting Tables, CSS for presentation, linking CSS to a web pages and three basic page layout strategies. JavaScript provides a way to validate form's data on the client's computer. Students are requested to design and implement professional website regarding skills of the contents of the topic.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	At the end of the course, students are expected to learn: 1- Understanding the web design development project lifecycle. 2- Identify HTML and create a basic page with these languages. 3- Identify the types of images used in modern web design and explain what types are appropriate for different functions. 4- Designing and programming to create all required aspects of website, controlling on text, tables, hyperlinks, colors as well as using selected templates to design website. The student will write professional coding to create the above elements. 5- Identify the CSS for presentation and add interactivity to website by JavaScript.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I: [8 hrs.] Understanding of web design and programming Concepts: Students should gain a comprehensive understanding of the fundamental concepts of web page, web site, web operations, web browser, Uniform Resource Locator (URL) and internet protocols UNIT-II: [12 hrs.] Web design pages structure: Web design typically follows a structured approach to create organized and efficient web pages such heading and paragraph using tags and

	attributes concepts. Then lists are used to group related items by using different lists types. UNIT-III: [20 hrs.] Using images and Links in HTML page: In this section the different types of images with different source are embedded in HTML pages. Then different links types (external, internal, directory and hyperlinking) are used to provide coherence and freedom of movement between different pages in the designing web. UNIT-IV: [20 hrs.] Tables and Forms: In this section the tables with rows and columns spanning are embedded in web pages. Then design a form with different elements types are used to build a web project. All concepts used in units (I, II and III) are included in forms. Also, Text, Checkbox, Password and others are included in the web site. UNIT V [20 hrs.] CSS for presentation and JavaScript Language: The CSS are used to make the web page more attractive and adding the CSS format for each HTML element. The JavaScript language is used to add interactivity, enhance user experience, and create dynamic web page such as providing immediate feedback, handle events efficiently, and making applications more responsive.
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<u>Learning Strategies:</u> The best way to learn a web design programming is to practice them. You can also find books, courses, or tutorials that offer exercises and examples. 1- Hands-on Practice: web design programming require a continues practice and working on building simple systems and developing them gradually contributes to the development of practical practice 2- Continuous reading of the latest developments in the subject of website design helps in developing skills and thus increasing experience in the subject of design. 3- Collaborative Learning by participating and collaborating with a team in building a web site project that includes a number of web pages with joining collaborative groups helps develop skills and solve problems collectively <u>Teaching Strategies:</u>

	1- Conceptual Explanation: start by providing a simple explanation of web design programming with simple examples of limited web pages then combines these pages into web site. 2- Hands-on Exercises: It includes distributing students to a number of projects (two students for each project) and following up on the development of the projects with the theoretical and practical material in the course. 3- Interactive Sessions: Conduct live sessions where students can actively participate, ask questions, and solve problems. This approach promotes engagement and practical learning. 4- Resources and References: Share relevant web design 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	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	46	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	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)	6 and 10	LO #1, #2, #3, #4, #5
	Assignments	2	8% (8)	4 and 11	LO #1, #2, #3, #4, #5
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3, #4, #5
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to web-design: Internet Technology, the Web Concepts, Uniform Resource Locator (URL) and Internet Protocols.
Week 2	Web Design: Client-Server Architecture, Web Browsers, Web Operations, Site Structure Types of Web Sites, Website Design Issues
Week 3	HTML Document Structure: HTML Structure, HTML Tags and Attributes, HTML Headings, HTML Paragraphs, HTML Character Entities
Week 4	HTML List: Html List Advantages, List Types, Unordered Lists, Ordered Lists, Description Lists List Type Selection, Nesting Lists
Week 5	Images in HTML: Introduction to Image Types, Image Attributes, Adding Images
Week 6	Links in HTML: Linking in HTML, Relative and Absolute Path, Target attribute, External linking Internal linking, Directory Linking, Hyperlinking to other Content
Week 7	Midterm Exam1 /
Week 8	Tables in HTML: Introduction, Table Element, Adding Tables, Spanning Cells
Week 9	Topic in Web Design Forms: Form Elements Types, Forms Attributes, Form Controls, Input Element Type Attribute (Text, Checkbox, Password), Fieldset and Select Element, Form Design
Week 10	CSS for Presentation: Fundamental and Advantage of CSS, Types of CSS (Inline, Embedded and External CSS), Linking CSS to a Web Page, Grouped and ID Selectors
Week 11	CSS for Presentation: Formatting Text, Formatting Lists, Colors and Backgrounds, Styling Tables
Week 12	Page Layout with CSS: Page Layout Strategies, Fixed Layout, Fluid Layout, Elastic Layout, Hybrid Layout
Week 13	Midterm Exam2 /
Week 14	JavaScript: Basic Scripting: Adding JavaScript to a Page, Accessing Nodes, Event Driven
Week 15	JavaScript: Basic Handling: Form Validation, Regular Expressions, Basic Validation, Data Format Validation
Week 16	Final exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to HTML: to understand the student the general idea of the concept of HTML, open and save file of HTML.
Week 2	Basic and concept of HTML markup: practice the student about the structure of HTML markup, bigining, ends and main parts of HTML markup
Week 3	HTML markup with examples: Learn the students Tags and Attributes HTML Headings, Paragraphs, Character Entities
Week 4	HTML markup with list: learn and practice student all types of HTML list (order, an order, nested and description lists)

Week 5	HTML markup with images: learn student how to load image in HTML markup
Week 6	HTML markup with different types of linking: learn the student to use different types of links (internal, external Directory Linking and Hyperlinking) and using links with images.
Week 7	Midterm Exam
Week 8	HTML markup with table: learn the student to create table, spanning tables in row and column.
Week 9	HTML markup with form: learn the students to design the form with different element types, attributes and controls, Fieldset and Select Element.
Week 10	Design project: apply a practical project using HTML markup.
Week 11	HTML markup with CSS: using different types of CSS for Presentation (Inline, Embedded and External CSS), using CSS in web pages.
Week 12	HTML markup with CSS (images, tables, lists and forms).
Week 13	HTML markup with CSS (Contextual Selector): using the different page Layout (fixed, elastic and hybrid) with CSS.
Week 14	Introduction to JavaScript Language: Adding JavaScript to an HTML page
Week 15	Project Discussion

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	- Learning Web Design, 4th Edition, Jennifer Niederst Robbins. - HTML & CSS: The Complete Reference, Fifth Edition, Thomas A. Powell.	
Recommended Texts	HTML and CSS Design and Build Websites, Jon DuCkeTT.	
Websites	Designing for the Web – free (Online Edition) - http://designingfortheweb.co.uk . W3Schools Online Web Tutorials http://www.w3school.com .	

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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Software Engineering		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010602106			
ECTS Credits	4			
SWL (hr/sem)	100			
Module Level	UGII	Semester of Delivery		
Administering Department	Computer Science	College	College of Science	
Module Leader	Dr. Huda Haikal Hussein		e-mail	huda.atmsc@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification		
Module Tutor	Lecturer Zeyad Farooq Lutfi		e-mail	Zeyadfa6@uomustansiriyah.edu.iq
Peer Reviewer Name	D. Hudi Abdulalli		e-mail	
Scientific Committee Approval Date	25 / 4 /2024		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	This course covers the fundamentals of software engineering, including system requirements, effective methods of design, and testing, teams' software development, and the application of engineering tools. Also, the course will initiate students to the different software process models, software requirements engineering process, systems analysis and design as a problem-solving activity, key elements of analysis and design, phases within the system development life.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completion of the course, the students will be able to: 1. Explain the introduction of software engineering practices that are key components. 2. Identify methods that will lead to the creation of a software architecture that achieves a specified level of reliability, dependability and security. 3. Apply the basic principles of software project management in a team environment and introduce different application types. 4. Introduce different case studies for more explanation. 5. Understand a variety of strategies to the software process models and explain their benefits and problems. 6. Understand a variety of strategies to the testing of simple programs. 7. Identify the principal issues associated with software evolution and explain their impact on the software lifecycle. 8. Discuss the system requirement type: functional and non-functional requirements.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I: [15 hrs] - Introduction to Software Engineering, Software costs, Software products, Product specification, and frequently asked questions about Software Engineering. - Essential attributes of good software, Importance of software engineering, Software process activities, Application types, Software engineering fundamentals, and Web-based software engineering. - Case studies, A personal insulin pump, A mental health case patient management system, and A wilderness weather station. UNIT-II: [16 hrs] - Software Processes, Software process descriptions, Plan-driven and agile processes, and Software process models. Iterative Waterfall Model, Incremental

	development, Reuse-oriented software engineering, Incremental development benefits, Incremental development problems. Process activities, the requirements engineering process, Software design and implementation, A general model of the design process, Requirements Analysis and Specification. UNIT-III : [12 hrs] Rapid software development, Agile methods, Agile manifesto, The principles of agile. UNIT IIII: [20 hrs] - Requirements Engineering, what is a requirement? Types of requirements, User and system requirements. System stakeholders, Stakeholders in the Mentcare system. Functional and non-functional requirements, Mentcare system: functional requirements, Types of nonfunctional requirement, Non-functional requirements implementation, Non-functional classifications.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<u>Learning Strategies:</u> The best way to learn and explain software engineering models and discuss the differences among them with their benefits and problems. You can also find books, courses, or tutorials that offer case studies for development system. 1- software engineering models Reading and Analysis: Study and analyze existing model examples and systems written by experienced developers. Understand. 2- Collaborative Learning: Engage in discussions and collaborations with fellow learners or join programming communities and forums to share knowledge, ask questions, and solve problems collectively. <u>Teaching Strategies:</u> 1- Conceptual Explanation: Start by providing clear and concise explanations of software engineering concepts. Use real-life examples for case studies to help students grasp abstract concepts easily.

	2- Hands-on Coding Exercises: Include assignments that reinforce the concepts taught for types of software engineering models. Give examples for using use case models to identify system requirements. Encourage students to apply the model into practice, helping them develop their understanding skill. 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 software engineering design and analysis 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	4	12% (12)	3, 7, 9 and 12	LO #1, #2, #3, #4, #5
	Assignments	2	10% (10)	5 and 14	LO #5, #6, #7, #8
	Seminar/	1	10% (10)	Continuous	All
	Report	1	8% (8)	Continuous	All
Summative assessment	Midterm Exam	1hr	10 % (10)	7	LO #1, #2, #3, #4, #5
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Software Engineering, Software costs, Software products, Product specification, frequently asked questions about Software Engineering.
Week 2	Essential attributes of good software, Importance of software engineering, Software process activities, Application types, Software engineering fundamentals, Web-based software engineering.
Week 3	Case studies, A personal insulin pump, A mental health case patient management system, and A wilderness weather station.
Week 4	Software Processes, Software process descriptions, Plan-driven and agile processes, and Software process models.
Week 5	Iterative Waterfall Model, Incremental development, Reuse-oriented software engineering, Incremental development benefits, Incremental development problems.
Week 6	Process activities, the requirements engineering process, Software design and implementation, A general model of the design process, Requirements Analysis and Specification.
Week 7	Midterm Exam /
Week 8	Software Analysis & Design Tools, Structured Design, Software evolution, System evolution.
Week 9	Rapid software development, Agile methods, Agile manifesto, The principles of agile.
Week 10	Requirements Engineering, what is a requirement? Types of requirements, User and system requirements.
Week 11	System stakeholders, Stakeholders in the Mentcare system.
Week 12	Functional and non-functional requirements, Mentcare system: functional requirements, Types of nonfunctional requirement, Non-functional requirements implementation, Non-functional classifications.
Week 13	Requirements discovery, Interviewing, Interviews in practice, Problems with interviews.
Week 14	Stories and scenarios, Scenarios, Use cases, Use cases for the Mentcare system.
Week 15	Seminar
Week 16	Preparatory week before the Final Exam

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	
Week 12	
Week 13	
Week 14	
Week 15	

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	[1]: Sommerville, Ian. Software engineering / Ian Sommerville. — 9th ed & 10th. ISBN 10: 0-13-703515- 2, ISBN 13: 978-0-13-703515-1. Publisher: Copyright © 2011, 2006, 2005, 2001, 1996 Pearson Education, Inc., publishing as Addison-Wesley.	
Recommended Texts	[1]: Mall Rajib, Fundamentals of Software Engineering, PHI. [2]: Pressman, Software Engineering Practitioner's Approach, TMH.	
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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	Algorithm Design and Analysis		Module Delivery	
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU010602201			
ECTS Credits	6			
SWL (hr/sem)	150			
Module Level	UGII	Semester of Delivery		4
Administering Department	Computer Science	College	College of Science	
Module Leader	Dr. Bassam Basim Jamil		e-mail	Dr.balkindy@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant. Prof		Module Leader's Qualification	
Module Tutor	Assistant.Prof Nada Thanoon Ahmed		e-mail	nada_thanoon@uomustansiriyah.edu.iq
Peer Reviewer Name	Assistant.Prof Nada Thanoon Ahmed		e-mail	nada_thanoon@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	Programming Fundamentals (I)	Semester	1
	Programming Fundamentals (II)	Semester	2
	Data Structures	Semester	3
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 covers the fundamentals of algorithmic problem solving, fundamentals of analysis of algorithm efficiency, studying set of important algorithm in different areas of computing, space and time trade-offs .concentration on algorithms applications and development of algorithms. Also the course will initiate students to the different requirements algorithms analysis and design as a problem-solving activity, key elements of analysis and design. This course will cover the basic approaches and mindsets for analyzing and designing algorithms. Topics include the following: Worst and average case analysis. Recurrences and asymptotic. Efficient algorithms for sorting and searching. Algorithm design techniques: divide-and-conquer, dynamic programming, greedy algorithms. Algorithms for fundamental graph problems .
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon completion of the course, the students will be able to: 1- Understand a set of important algorithms in different areas of computing. 2- The students should be able to design new algorithms and analyze their efficiency. 3- To provide further practice with programming and have the ability of building computer programs for general problems in different areas. 4- Developing their analytical skills. 5- Use Divide and Conquer algorithm to solve an appropriate problem. 6- Use Greedy Method and Dynamic programming to solve an appropriate problem.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I : [13 hrs] Introduction to Algorithms concepts , Best case Analysis, Worst Case Analysis, Average case Analysis. Complexity Notation : Big O notation , Omega Notation ,Theta Notation, Recursion Analysis. UNIT-II : [18 hrs] Sorting Algorithms: Simple iteration Sort ,Bubble Sort Algorithm , Selection Sort Algorithm , Insertion Sort Algorithm , Shell Sort Algorithm , Radix Sort Algorithm. UNIT-III : [19 hrs] Divide and conquer : Binary Search Algorithm , Quick Sort Algorithm , Merge Sort Algorithm, Heap Sort Algorithm. UNIT IIII : [23 hrs] Optimization in Algorithms Design Equations: Graph theory Greedy Algorithms: shortest paths (Dijkstra Algorithm), Knapsack problem Dynamic Programming :

	-Multistage graph problem: Forward Approach, -Multistage graph problem: Backward Approach.
--	---

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<u>Learning Strategies:</u> The best way to learn and remember algorithms Design and analysis is to practice them. 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 Algorithms exercises , small projects, and problem-solving activities. 2- Algorithm Reading and Analysis: Study and analyze existing algorithm examples and programs written by experienced developers. Understand the algorithm and analysis , and write the program enhance your understanding of algorithm design. 3- Collaborative Learning: Engage in discussions and collaborations with fellow learners or join programming communities and forums to share knowledge, ask questions, and solve problems collectively.
	<u>Teaching Strategies:</u> 1- Conceptual Explanation: Start by providing clear and concise explanations of Algorithm design and analysis concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2- Hands-on Coding Exercises: Include practical Algorithm design exercises and assignments that reinforce the concepts taught. convert this algorithms to programs .Encourage students to apply the algorithm into practice, helping them develop their coding skill. 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 algorithm design and analysis 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	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	71	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	150		

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, #3, #4, #5
	Assignments	2	8% (8)	4 and 11	LO #1, #2, #3, #4, #5
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3, #4, #5.#6
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Algorithms concepts
Week 2	Best case Analysis, Worst Case Analysis, Average case Analysis. Complexity Notation : Big O notation , Omega Notation ,Theta Notation
Week 3	Recursion Analysis
Week 4	Sorting Algorithms: 1. Introduction

	2. Simple iteration Sort 3. Bubble Sort 4. Selection Sort
Week 5	Sorting Algorithms: 1. Insertion Sort 2. Shell Sort
Week 6	Sorting Algorithms: Radix Sort
Week 7	Midterm Exam / Divide and conquer : Binary Search
Week 8	Divide and conquer : Quick Sort ,
Week 9	Merge Sort
Week 10	Heap Sort
Week 11	Optimization in Algorithms Design Equations: Graph theory
Week 12	Greedy Algorithms: shortest paths (Dijkstra Algorithm)
Week 13	Greedy Algorithms: Knapsack problem
Week 14	Dynamic Programming : Multistage graph problem: Forward Approach
Week 15	Dynamic Programming : Multistage graph problem: Backward Approach.
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	power ,sum ,factorial program without recursion
Week 2	power ,sum ,factorial program with recursion
Week 3	Apply Simple iteration Sort ,
Week 4	Apply bubble sort and selection sort
Week 5	Apply insertion sort
Week 6	Apply shell sort
Week 7	Midterm Exam / Apply radix sort
Week 8	Apply Binary Search
Week 9	Apply Quick Sort
Week 10	Apply Merge Sort
Week 11	Apply Heap Sort
Week 12	Dijkstra Algorithm

Week 13	Knapsack problem
Week 14	Multistage graph problem Forward Approach
Week 15	Multistage graph problem: Backward Approach.

Learning and Teaching Resources		
مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	[1]: Harsh Bhasin, Department of Computer Science Jamia Hamdard New Delhi, ALGORITHM Design and Analysis© Oxford University Press 2015.	
Recommended Texts	[1]: Anany Levitin, "Introduction to the Design and Analysis of Algorithms " , Tsinghua University Press, 2013.	
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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Database Fundamentals		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	MU010602202		
ECTS Credits	6		
SWL (hr/sem)	150		
Module Level	UGII	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader	Dr. Ismael Abdulsattar Jabbar	e-mail	ismaelabdul@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant. Prof	Module Leader's Qualification	
Module Tutor	Dr. Jalal Hameed Awad	e-mail	jalalhameed@uomustansiriyah.edu.iq
Peer Reviewer Name	Dr. Jalal Hameed Awad	e-mail	jalalhameed@uomustansiriyah.edu.iq
Scientific Committee Approval Date	15 / 5 /2024	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	This course presents the following: 1- fundamentals detail of designing database system. 2- Build a different database system module. 3- formalizing the structured of the queries. 4- Normalization of relations, data integrity, database security, and database administration. 5 - develop a project based on the knowledge they gained.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Learn the database fundamentals systems, that means they will be able to: 1. Produce a conceptual design of a database using E/R modelling. 2. □Convert the conceptual design to a logical design using the relational model. 3. Express queries to the relational model using relational algebra, relational calculus, and SQL. 4. Perform normalization as necessary for good relational database design. 5. Construct a database schema, enter data, perform updates, and query a database using SQL server. 6. □Produce a formal document for database systems project. 7. □Give a formal presentation on a database systems project

Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I : [8 hrs] Introduction to Database, How to choose the Database for holding and processing types of data , Memory Representation ,Introduction to Abstract Data Type , Introduction to Time Complexity (Space Complexity, Time Complexity) UNIT-II : [22 hrs.] File systems vs. DBMS Approaches to and evolution of DBMS Database architecture and data independence Core DBMS functions and system components DBMS user, designer, application developer, administrator UNIT-III: [28 hrs.] □ Conceptual modeling – ER model □ Logical database design □ Relational model, □ Object relational model □ Relational database design, schema design, mapping □ Conceptual schema to relational schema, functional □ Dependencies, super keys and candidate keys, foreign □ Keys, schema decomposition and refinement, loss-less □ Join and dependency preservation, normal forms – 1NF, □ 2NF, 3NF, BCNF, multi-valued dependency and 4NF –
	□ entity and referential integrity □ UNIT III: [15 hrs.] □ Relational algebra □ Relational calculus □ SQL □ SQL queries, constraints and triggers □ ODBC, JDBC □ Query processing strategies □ Query evaluation □ Query processing costs

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	<u>Learning Strategies:</u> The best way to learn and remember database systems is to practice them. 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 Database exercises, small projects, and problem-solving activities. 2- Reading types of database: Study and analyze existing database examples and programs written by experienced developers. write the programs enhance your understanding of Database. 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. <u>Teaching Strategies:</u> 1- Conceptual Explanation: Start by providing clear and concise explanations of Database concepts. Use real-life examples or analogies to help students grasp abstract concepts easily. 2-Hands-on Coding Exercises: Include practical Database exercises and assignments that reinforce the concepts taught.
	3-Convert the types of Database system to programs. Encourage students to apply the algorithm into practice, helping them develop their coding skill. 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 Database systems 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	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	71	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	150		

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, #3, #4, #5
	Assignments	2	8% (8)	4 and 12	LO #1, #2, #3, #4, #5
	Projects / Lab.	1	14% (14)	Continuous	LO #3, #4, #5, #6, #7
	Report	1	8% (8)		LO #1, #2, #3, #4, #5
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4, #5
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المناهج الأسبوعي النظري

	Material Covered
Week 1	Introduction to the course Structure, Curriculum, Assessment, Development Environment, Equipment, Independent Learning.
Week 2	Introduction to database systems What is database, tables, file systems vs database systems, uses of database, what is DBMS.
Week 3	Introduction to the database concepts Data, information, knowledge and wisdom. Database concepts

Week 4	Introduction to the database environment Definition of DBMS, the role of DBMS, database types, database classifications, data independence
Week 5	Database design and data modelling Database development life cycle Introduction to Hierarchical, network, relational and object-oriented models.
Week 6	Relational database model Relational data, keys, integrity rules
Week 7	Midterm Exam
Week 8	Relationships with relational data. Relational set operators
Week 9	Entity relationship (ER) modelling
Week 10	Entities, attributes and relationships Developing an ER diagram
Week 11	Normalization of database tables What normalization is and what role it plays in the database
Week 12	Design process. About the normal forms 1NF, 2NF, 3NF, BCNF, and 4NF
Week 13	Utilizing the DBMS process on applied problem
Week 14	Catch-up week Review materials covered so far, Q&A Review delivery so far
Week 15	Exam 2
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Introduction to SQL language
Week 2	What is DDL & DML
Week 3	Relations/primary key& foreign key
Week 4	Retrieval queries in SQL
Week 5	WHERE clause with Quiz
Week 6	(Between) & (IN) & (NOT) operators
Week 7	Midterm Exam
Week 8	LIKE operator
Week 9	Logical operators (AND/OR) and (Ambiguous attributes names)
Week 10	ORDER BY instructions

Week 11	AS, DISTINCT clauses With quiz
Week 12	Aggregate functions
Week 13	Aggregate functions(cont.)
Week 14	GROUPING clause HAVING clause
Week 15	Action quires: INSERT,DELETE, UPDATE

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	[1]: Elmasri, R. and Navathe, S.B., 2016. Fundamentals of Database Systems 7th ed.	Yes
Recommended Texts	[1]: Coronel, C. and Morris, S., 2019. <i>Database systems: design, implementation and management</i> . Cengage learning.	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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Compilers		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	MU010602203		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	UGII	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader		e-mail	
Module Leader's Acad. Title	lecturer	Module Leader's Qualification	
Module Tutor		e-mail	
Peer Reviewer Name		e-mail	
Scientific Committee Approval Date	22 / 4 /2024	Version Number	1.0

Relation with other Modules			
العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Theory of Computation (2)	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	The module objectives of a Compiler course typically cover the fundamental principles, techniques, and tools involved in the design and implementation of a compiler. Below are some common objectives you might find in such a course: Understanding the Role of Compilers, explain the role of a compiler in the software development process. Differentiate between various types of compilers and interpreters. Understand the phases of compilation: lexical analysis, syntax analysis, semantic analysis, optimization, and code generation. Practical Compiler Implementation Apply the theoretical knowledge to implement a simple working compiler for a small programming language. Test and debug the compiler to ensure it correctly translates source code into executable code. Explore the challenges and considerations in real-world compiler design. These objectives aim to equip students with both the theoretical understanding and practical skills needed to design and build a basic compiler.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	<i>At the end of the course, students are expected to learn:</i> • Students will have the experience of designing and constructing compiler. • Developing the Students ability to design and implement compilers for specific purposes. • Acquire familiarity with basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines.
Indicative Contents المحتويات الإرشادية	The indicative contents of a Compiler course typically encompass a range of topics that cover the theoretical foundations, practical techniques, and tools necessary for understanding and building compilers. Below is an outline of the typical content: UNIT I: [13 hrs.] Introduction to Compilers: Overview of Compilers Definition, functions, and importance. History of compiler development. Difference between compilers and interpreters. Structure of a typical compiler: Frontend, middle-end, and backend. Compilation process and its phases. UNIT-II: [18 hrs.] Lexical Analysis: Role of the lexical analyzer (lexer). Tokens, lexemes, and patterns. Regular expressions and finite automata. Construction of lexical analyzers. Transition diagrams, state machines. Tools for lexical analysis: Lex/Flex. Handling errors in lexical analysis. UNIT-III: [19 hrs.] Syntax Analysis: Role of the syntax analyzer (parser). Context-free grammars (CFGs).

	Parse trees and abstract syntax trees (ASTs). Parsing techniques: Top-down parsing (e.g., Recursive Descent, LL parsing), and Bottom-up parsing (e.g., LR parsing, SLR, LALR). Construction of parsers: Parser generators like Yacc/Bison. Error handling and recovery in parsing. UNIT-IV: [10 hrs.] Semantic Analysis: Purpose of semantic analysis. Symbol tables: Construction and management. Type checking and type systems. Attribute grammars and syntax-directed translation. Semantic error detection and reporting. UNIT V (12) Intermediate Code Generation: Three-address code (TAC), Quadruples, Triples Code Optimization: Local vs. global optimizations. Data flow analysis. optimization techniques. Code Generation. Error Handling and Recovery
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	<u>Learning Strategies:</u> Learning strategies for a Compiler course are essential for mastering the complex concepts and practical skills required to design and implement compilers. Here are some effective strategies: 1. Understand the Big Picture • Conceptual Framework: Start by understanding the overall structure of a compiler and how each phase (lexical analysis, syntax analysis, semantic analysis, etc.) fits into the larger process. • Flow of Compilation: Map out the flow from source code to executable code, identifying the role of each component. 2. Start with Theoretical Foundations • Study Formal Languages and Automata: Gain a solid grasp of regular expressions, context-free grammars, and finite automata, as these form the basis of lexical and syntax analysis. • Mathematical Models: Engage with the mathematical models behind parsing techniques and type checking. 3. Hands-on Practice • Build Simple Components: Begin by implementing basic components like a lexical analyzer and gradually move on to more complex ones like parsers and code generators.

- **Use Compiler Construction Tools:** Get familiar with tools like Lex and Yacc or ANTLR. Start with small projects to see how these tools automate parts of the compilation process.

4. Incremental Learning

- **Phase-by-Phase Implementation:** Focus on implementing one phase of the compiler at a time. For example, start with the lexer, then move on to the parser, semantic analyzer, and so on.
- **Iterative Testing:** Continuously test each phase with small examples to ensure correctness before moving on to the next phase.

5. Collaborative Learning

- **Study Groups:** Join or form study groups to discuss concepts, solve problems, and share insights. Explaining concepts to peers is a powerful way to reinforce your understanding.
- **Pair Programming:** Work with a partner on coding assignments to learn from each other's strengths and different problem-solving approaches.

6. Visualization Techniques

- **Draw Parse Trees and Automata:** Visualize how parsers and finite automata work by drawing parse trees and state machines for different inputs.
- **Diagram Compilation Phases:** Use flowcharts and diagrams to represent the flow of data and control between different phases of the compiler.

7. Apply Concepts to Real-world Languages

- **Case Studies:** Analyze the compilers of existing programming languages like C, Java, or Python to see how theoretical concepts are applied in real-world scenarios.
- **Dissect Open-source Compilers:** Explore open-source compiler projects like GCC, LLVM, or the Java Compiler to understand how professional-grade compilers are built.

8. Iterative Project Work

- **Start Small:** Begin with simple projects like building a calculator or a basic interpreter for a mini-language.
- **Extend Projects:** Gradually add features like type checking, optimization, or code generation to your projects, reflecting the stages of a full-fledged compiler.
- **Capstone Project:** Work on a comprehensive project where you build a compiler for a simple programming language from scratch.

9. Seek Feedback and Continuous Improvement

- **Code Reviews:** Regularly seek feedback on your code from instructors or peers. Participate in code reviews to learn best practices and optimization techniques.
- **Refactor Code:** Revisit and refactor your code as you learn more advanced techniques to make it more efficient and maintainable.

10. Use Online Resources and Documentation

- **Compiler Textbooks:** Refer to standard textbooks like "Compilers: Principles, Techniques, and Tools" (the Dragon Book) for in-depth understanding.
- **Online Tutorials and Courses:** Leverage MOOCs, video lectures, and online tutorials that break down complex topics into manageable segments.
- **Documentation:** Frequently refer to documentation for tools like Lex, Yacc, or LLVM to understand their capabilities and limitations.

11. Practice Problem-Solving

- **Work on Compiler Design Problems:** Solve problems related to grammar design, syntax tree construction, and optimization techniques.
- **Participate in Competitions:** Engage in coding competitions or challenges focused on compilers and language design to apply your knowledge in a competitive environment.

12. Reflect and Revise

- **Regular Reviews:** Periodically review your notes and revisit key concepts to reinforce your understanding.
- **Self-assessment:** Test your knowledge with quizzes, exercises, and by trying to explain concepts without referencing your notes.

These learning strategies will help you build a strong foundation in compiler design, enabling you to handle both theoretical and practical challenges.

Teaching Strategies:

Teaching a Compiler course effectively requires a combination of strategies that address both the theoretical concepts and practical skills involved. Here are some teaching strategies tailored for a Compiler course:

1. Start with a Conceptual Overview

- **Big Picture Introduction:** Begin the course by providing an overview of the entire compilation process. Explain the role of each

phase (lexical analysis, syntax analysis, semantic analysis, etc.) and how they work together to translate source code into machine code.

- Relate to Real-world Examples: Use familiar programming languages and their compilers (e.g., Java, C, Python) as examples to illustrate how compilers work in real-world applications.

2. Sequential and Incremental Learning

- Phase-by-Phase Approach: Teach the course in a sequence that mirrors the structure of a compiler: start with lexical analysis, then syntax analysis, semantic analysis, and so on. Each topic builds on the previous one, allowing students to see the progression from source code to executable code.
- Scaffold Learning: Begin with simple examples and gradually introduce more complexity. For instance, start with simple regular expressions before moving on to more complex lexical analysis tasks.

3. Hands-on Programming Exercises

- Implement Components: Assign programming exercises where students build individual components of a compiler, such as a lexer or parser. This helps them understand each phase in depth.
- Incremental Project Work: Have students start with a basic project (e.g., a simple calculator or interpreter) and add features incrementally, such as adding a syntax tree, type checking, or optimization.
- Use Compiler Tools: Introduce tools like Lex and Yacc early in the course, and provide assignments that require students to use these tools to generate parts of their compiler.

4. Active Learning Techniques

- Classroom Discussions and Debates: Encourage discussions on different parsing techniques, optimization strategies, or the design choices in well-known compilers. Debates can help students understand the trade-offs involved in compiler design.
- Peer Instruction: Use peer instruction methods where students explain concepts to each other. This can be particularly effective for complex topics like syntax-directed translation or register allocation.

5. Visualization and Diagrams

- Draw Parse Trees and Automata: Use diagrams to explain concepts like parse trees, finite automata, and control flow graphs. Visual representations help students grasp abstract concepts more easily.
- Flowcharts and Graphs: Use flowcharts to represent the data and

control flow within a compiler, and control flow graphs (CFGs) to explain optimization techniques.

6. Interactive and Adaptive Teaching

- **Use Interactive Tools:** Incorporate interactive tools or simulators that allow students to experiment with different phases of a compiler. For example, tools that visualize lexical analysis or parsing can make these concepts more tangible.
- **Adaptive Feedback:** Provide continuous feedback on assignments and projects. Adapt your teaching based on the common areas where students struggle, offering additional resources or explanations as needed.

7. Case Studies and Real-world Examples

- **Analyze Existing Compilers:** Use case studies of well-known compilers (like GCC or LLVM) to show how theoretical concepts are applied in practice. This can help bridge the gap between theory and real-world application.
- **Reverse Engineering:** Assign projects where students analyze or modify existing open-source compilers to understand how they are built and maintained.

8. Collaborative Learning

- **Group Projects:** Encourage students to work in groups on larger projects, such as implementing a compiler for a simple language. Collaborative projects can help students learn from each other and tackle more complex tasks than they could individually.
- **Code Reviews:** Organize peer code review sessions where students critique each other's compiler implementations, focusing on both correctness and optimization.

9. Integrate Theory with Practice

- **Theory-Driven Labs:** Follow up theoretical lectures with lab sessions where students implement the concepts discussed. For example, after teaching about LR parsing, have students implement an LR parser in a lab session.
- **Practical Examples:** During lectures, regularly switch between theoretical explanations and practical examples, such as showing how a regular expression is converted into a finite automaton, and then having students implement this in code.

10. Continuous Assessment and Feedback

- **Frequent Quizzes and Assignments:** Use short quizzes and frequent assignments to keep students engaged and to reinforce key

	concepts. • Iterative Project Feedback: Provide iterative feedback on project milestones, helping students correct mistakes and improve their understanding before the final submission.
	11. Guest Lectures and Industry Insights
	• Industry Experts: Invite guest speakers from industry to talk about real-world compiler development challenges, such as those faced in optimizing for modern processors or in just-in-time compilation. • Research Insights: Introduce students to current research trends in compiler design, such as machine learning-based optimizations, to inspire further learning and exploration.
	12. End-of-Course Capstone Project
	• Comprehensive Project: Conclude the course with a capstone project where students design and implement a simple compiler for a small programming language, integrating all the concepts learned throughout the course. • Project Presentation: Have students present their compiler projects, explaining the design choices, challenges faced, and solutions implemented. This reinforces their learning and allows for peer-to-peer learning. By combining these teaching strategies, you can create a well-rounded Compiler course that equips students with both the theoretical understanding and practical skills needed to design and build compilers.

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	46	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	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)	6 and 10	LO #1, #2, #3, #4, #5
	Assignments	2	8% (8)	4 and 11	LO #1, #2, #3, #4, #5
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		LO #1, #2, #3, #4, #5.#6
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1, #2, #3, #4
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to Translators: Compiler, Interpreter Assembler, Linker and Loader.
Week 2	Structure of Compiler: Analysis Phases, Construction Phases, Example of Compilation.
Week 3	Lexical Analyzer: Role of the Lexical Analyzer, Tasks of Lexical Analyzer, Input Buffering, Symbol Table
Week 4	Recognition of Tokens: Token, Lexeme, Pattern Specifications of Tokens, Operations on Languages
Week 5	Syntax Analysis (Parse): Role of the Parser, Interaction of lexical analyzer with parser, Parse Tree
Week 6	Context-Free Grammars (CFG): Ambiguity in Instructions, Regular Expression vs. CFG, Elimination of Left Recursion
Week 7	Midterm Exam
Week 8	Parsing Techniques: Top-Down Parser, Bottom-Up Parser, Compute of FIRST Function
Week 9	Top-Down Parsing: Backtracking, Predictive Parser, Design Parsing Table
Week 10	Bottom-Up Parsing: Shift-Reduce Parser, Operator-Precedence Parser
Week 11	Semantic analysis: Attributes and their computation, Type Checking, Semantic Errors
Week 12	Intermediate Code Generation: Control Structures, Three-Address Code, Operations Within Expressions
Week 13	Code Optimization: Introduction and View of Optimization, Global Optimization, Local Optimization.
Week 14	Code Generation: Issues in the Design of a Code Generator, Object Program, Assembly Controls
Week 15	Preparatory week before the final Exam

Week 16	Final exam
---------	------------

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Programming Languages: Types and Features of Programming language
Week 2	Types of Errors in source program, Errors Handling, Regular Expression
Week 3	Classification of source program: using Lexical Analyzer, Tables Management
Week 4	Transition Diagrams (TD), Lexical Errors. Assignment (1)
Week 5	Abstract syntax Tree, Derivation, Syntax Errors
Week 6	Context-Free Grammars (CFG): Elimination of Left Factoring, Assignment (2)
Week 7	Midterm Exam
Week 8	Parsing Techniques: Compute of FOLLOW Function
Week 9	Top-Down Parsing: LL(1) Grammar, Conditions of LL(1)
Week 10	Bottom-Up Parsing: Table Construction of OPP, Assignment (3)
Week 11	Semantic analysis: Run-Time Errors, Assignment (4)
Week 12	Intermediate Code Generation: Control Structures, Three-Address Code, Operations Within Expressions
Week 13	Code Generation: Simple Register Allocation, Assignment (5)
Week 14	Preparatory week before the final Exam
Week 15	Project Discussion

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	[1]: E Alfred V. Aho, RaviSethi and Jeffrey D. Ullman, "Compilers: Principles, techniques, & tools" (2nd edition), 2007. [2]: Torben Mogensen, "Introduction to Compiler Design," Springer-Verlag London, 2011.	
Recommended Texts	[1]: Torben Mogensen, "Basics of Compiler Design" DEPARTMENT OF COMPUTER SCIENCE UNIVERSITY OF COPENHAGEN, 2010 [2]: Adesh K. Pandey, "CONCEPTS OF COMPILER DESIGN" Second Edition:2011. [3]: Dilshad Hasan, "Compiler Design", Knowledge Book	

	Distributors, 1st Edition:2010.	
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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Python Programming		Module Delivery
Module Type	Core		☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar
Module Code	MU010602204		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	UGII	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader	Dr. Muhanad Tahrir Younis	e-mail	mtu@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Asst. Prof.	Module Leader's Qualification	
Module Tutor	DR. Ayman Majed	e-mail	aymanmajid@uomustansiriyah.edu.iq
Peer Reviewer Name	Lecturer Firas Ali Hashim	e-mail	firas70@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	Programming Fundamentals I (1) Programming Fundamentals II (1)	Semester	(1) (2)
Co-requisites module	None	Semester	

Module Aims, Learning Outcomes and Indicative Contents

أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Objectives أهداف المادة الدراسية	The Python Programming module is a comprehensive introduction to Python, one of the most popular and versatile programming languages in the world today. Throughout the course, students will be guided through the fundamentals of Python, starting from basic syntax and data types, progressing through more complex topics such as object-oriented programming (OOP), and culminating in advanced concepts like file handling, error handling, and working with libraries and modules. Each topic is introduced with clear explanations, practical examples, and hands-on coding exercises to reinforce understanding. Python's simplicity and readability make it an ideal first language for aspiring programmers, while its powerful capabilities have established it as a go-to language for professionals in various fields, including data science, artificial intelligence, and more. This course emphasizes practical problem-solving and real-world application, ensuring that students not only learn the syntax and structure of Python but also how to think computationally and develop efficient, maintainable code. The course also includes a series of projects that allow students to apply what they have learned to more substantial programming tasks. These projects range from simple command-line tools to more complex applications that involve data manipulation and interaction with external libraries. By the end of the course, students will have developed a solid foundation in Python programming and will be able to tackle intermediate and advanced Python tasks with confidence.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	By the end of the course, students will be able to: 1. Master Python Fundamentals: Understand the core syntax, data types, and control structures of Python. 2. Build Problem-Solving Skills: Develop the ability to break down problems and implement solutions using Python's functions, loops, and conditions. 3. Apply Object-Oriented Programming: Understand and apply object-oriented programming principles to create reusable and modular code. 4. Work with Data and Files: Efficiently handle data and file operations in Python. 5. Utilize Python Libraries: Leverage Python's standard libraries and third-party modules to extend functionality and perform data analysis using Python libraries such as Pandas and NumPy. 6. Develop Real-World Applications: Design and implement Python applications with graphical user interfaces. 7. Prepare for Advanced Learning: Develop a foundation for advanced topics in programming and software development.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. UNIT I: [5 hrs.] Introduction to Python: This unit introduces students to the fundamentals of programming and Python. It covers the basics of setting up the programming

environment, writing simple Python programs, and understanding the Python interpreter and syntax.

UNIT-II: [10 hrs.]

Python Basics: Variables, Data Types, and Operators: This unit covers Python's basic building blocks, including variables, data types, and operators. Students will learn how to handle different types of data and perform basic operations.

UNIT-III: [10 hrs.]

Control Structures: Conditional Statements and Loops: This unit introduces control flow in Python, including conditional statements and looping structures. Students will learn to manage the flow of their programs using if, else, while, and for statements.

UNIT-IV: [10 hrs.]

Functions and Modular Programming: This unit focuses on defining and using functions in Python. It covers function arguments, return values, and the concept of modular programming by creating and importing modules.

UNIT-V [10 hrs.]

Data Structures: Lists, Tuples, Dictionaries, and Sets: This unit explores Python's built-in data structures, including lists, tuples, dictionaries, and sets. Students will learn how to store, access, and manipulate data using these structures.

UNIT-VI [5 hrs.]

Working with Strings: This unit covers string manipulation and formatting in Python. Students will learn how to handle and process strings, including using regular expressions for pattern matching.

UNIT-VII [5 hrs.]

File Handling: This unit covers string manipulation and formatting in Python. Students will learn how to handle and process strings, including using regular expressions for pattern matching.

UNIT-VIII [10 hrs.]

Object-Oriented Programming (OOP): This unit introduces the concepts of object-oriented programming, including classes, objects, inheritance, polymorphism, and encapsulation. Students will learn to define classes, create objects, and use class attributes and methods.

UNIT-IX [5 hrs.]

Working with Libraries and Modules: This unit covers the use of Python libraries and modules, such as Numpy, Pandas, Matplotlib and Tkinter. Students will learn to use

	standard libraries, third-party libraries, and create custom modules for extended functionality.
--	--

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The following Learning and Teaching strategies are designed to create an engaging, interactive, and supportive learning environment that caters to different learning styles while ensuring that students gain a deep and practical understanding of Python programming.
	Learning and Teaching Strategies Lectures and Demonstrations Description: Each session will begin with a lecture introducing the key concepts and theoretical aspects of the topic at hand. Lectures will be supported by live coding demonstrations, where the instructor will walk through examples to illustrate how the concepts are applied in practice. Objective: To provide a clear and structured understanding of Python programming concepts and their real-world applications. Hands-On Coding Exercises Description: After each lecture, students will engage in hands-on coding exercises designed to reinforce the material covered. These exercises will range from basic syntax practice to more complex problem-solving tasks. Objective: To ensure students apply what they have learned immediately, reinforcing their understanding and building confidence in their programming skills. Collaborative Learning and Group Projects Description: Students will work in small groups on select assignments and projects, encouraging collaboration and peer learning. Group projects will simulate real-world scenarios where teamwork and collaboration are essential. Objective: To develop teamwork and communication skills while deepening understanding through peer discussion and collaboration. Interactive Labs Description: Dedicated lab sessions will be provided where students can experiment with Python programming in a guided environment. The labs will focus on specific tasks, such as debugging, file handling, or working with libraries. Objective: To offer students a space for experimentation and deeper exploration of complex topics, with instructor support available for troubleshooting and guidance. Regular Quizzes and Assessments Description: Quizzes will be administered regularly to assess students' understanding of the material. These assessments will include both theoretical questions and practical coding tasks. Objective: To track progress and identify areas where students may need additional support, ensuring continuous learning and improvement.

	6. Final Project ○ Description: The course will culminate in a final project where students will develop a complete Python application. This project will require the integration of all the concepts learned throughout the course. ○ Objective: To provide a capstone experience where students demonstrate their ability to apply Python programming skills to a real-world problem, from planning and development to testing and presentation. 7. Flipped Classroom Approach ○ Description: For certain topics, a flipped classroom approach will be used, where students are provided with reading materials, videos, or tutorials to review before class. Classroom time will then be used for discussion, problem-solving, and hands-on activities related to the pre-class materials. ○ Objective: To maximize classroom interaction and hands-on learning, while encouraging students to engage with the material at their own pace before applying it in class. 8. Peer Review and Code Review Sessions ○ Description: Students will participate in peer review sessions where they will critique and provide feedback on each other's code. These sessions will also include guided code reviews by the instructor to highlight best practices and common pitfalls. ○ Objective: To develop critical thinking and attention to detail, as well as to promote best practices in coding, such as readability, efficiency, and maintainability. 9. Use of Online Resources and Tools ○ Description: Students will be encouraged to use online platforms like GitHub for version control, Stack Overflow for problem-solving, and Python documentation for reference. Additionally, interactive coding platforms like Jupyter Notebooks may be used for in-class activities. ○ Objective: To familiarize students with the tools and resources used by professional Python developers, fostering independent learning and resourcefulness. 10. Regular Feedback and One-on-One Sessions ○ Description: The instructor will provide regular feedback on assignments and projects, and one-on-one sessions will be available for students who need extra help or have specific questions. ○ Objective: To ensure personalized support for each student, addressing individual challenges and helping them achieve their learning goals.
--	---

Student Workload (SWL)			
الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	79	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem)	46	Unstructured SWL (h/w)	3

الحمل الدراسي غير المنتظم للطالب خلال الفصل	الحمل الدراسي غير المنتظم للطالب أسبوعياً
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)	5 and 11	LO #1, #2
	Assignments	2	8% (8)	6 and 13	LO #1, #2, #3, #4
	Projects / Lab.	1	14% (14)	Continuous	All
	Report	1	8% (8)		All
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 الأسبوعي النظري	
	Material Covered
Week 1	Introduction to Python: Overview of Python: history, features, and applications, Installing Python and setting up the development environment, Writing and executing your first Python program, Understanding the Python interpreter and syntax basics
Week 2	Python Basics: Variables, Data Types, and Operators: Variables and assignment in Python Data types: integers, floats, strings, and Booleans, Basic operators: arithmetic.
Week 3	Python Basics: Variables, Data Types, and Operators: Basic operators: comparison, logical, and assignment, Type conversion and type checking, Input and output functions.
Week 4	Control Structures-Conditional Statements: if, if else, Nested if..else, if-elif-else, and Ternary Expression
Week 5	Control Structures- Loops: Looping structures: while and for loops, Nested Loop, Loop control statements: break, continue, and pass
Week 6	Functions and Modular Programming: Defining and calling functions, Function arguments: positional, keyword, and default values, Returning values from functions, Scope and lifetime of variables.
Week 7	Functions and Modular Programming: Modular programming: creating and importing modules, The concept of recursion.
Week 8	Midterm Exam

Week 9	Data Structures-Lists and Tuples: Lists: creation, indexing, slicing, and methods, Tuples: immutability and use cases.
Week 10	Data Structures- Dictionaries and Sets: Dictionaries: key-value pairs, methods, and practical applications, Sets: uniqueness, set operations, and methods, Iterating over data structures.
Week 11	Working with Strings: String creation and manipulation, String methods and operations, String formatting techniques, Handling special characters and escape sequences, Regular expressions for pattern matching
Week 12	File Handling: File operations: opening, reading, writing, and closing files, Working with text and binary files, File modes: read, write, and append, Handling file exceptions, Reading and writing CSV files.
Week 13	Object-Oriented Programming (OOP): Introduction to OOP concepts: classes and objects, Defining classes and creating objects, Class attributes and methods, The self keyword and object state.
Week 14	Working with Libraries and Modules: Introduction to Python standard libraries, Importing modules and using built-in functions, Working with third-party libraries (e.g., NumPy, Pandas, Matplotlib and Tkinter), Creating basic windows and dialogs, Adding widgets: buttons, labels, text boxes, etc.
Week 15	Project Discussion
Week 16	Final exam

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Setting up Python on different operating systems (Windows, macOS, Linux) and writing a simple "Hello, World!" program. Exploring the Python interpreter through basic commands.
Week 2	Lab 2: Writing programs to explore variables, data types, and operators.
Week 3	Lab 3: Creating small programs to perform calculations and handle user input/output.
Week 4	Lab 4: Developing programs to implement conditional statements, such as calculating max number, checking odd and even numbers, and creating a basic calculator.
Week 5	Lab 5: Developing programs to implement loops, such as calculating factorials, checking prime numbers, and creating patterns.
Week 6	Lab 6: Writing functions to solve specific tasks, such as max number and sum.
Week 7	Lab 7: Writing functions to solve recursion tasks, and breaking down larger programs into smaller modules. Example: Implementing a recursive function to calculate sum, factorial, power, and Fibonacci series.
Week 8	Midterm Exam
Week 9	Lab 8: Implementing programs that manipulate Lists and Tuples data structures. Examples include sorting a list, checking whether an element exists within a tuple, and removing an item from a list and a tuple.

Week 10	Lab 9: Implementing programs that manipulate Dictionaries and Sets data structures. Examples include finding the intersection of sets, and using dictionaries to count word frequencies in a text.
Week 11	Lab 10: Building programs that perform string manipulation and pattern matching using regular expressions. Example: Validating email addresses or parsing text files for specific patterns.
Week 12	Lab 11: Creating programs to read from and write to files. Handling exceptions related to file operations. Example: Writing a program to read data from a CSV file, process it, and write the output to another file.
Week 13	Lab 12: Creating basic classes and objects. Implementing simple applications using OOP concepts such as a class for a BankAccount with methods to deposit and withdraw money.
Week 14	Lab 13: Developing programs that utilize various libraries and modules. Example: Using the Pandas library to manipulate and analyze datasets, or using Matplotlib for basic plotting. Building a basic GUI application using Tkinter. Example: Developing a simple calculator or a to-do list application with a graphical user interface.
Week 15	Project Discussion
Week 16	Final exam

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	Matthes, Eric. <i>Python crash course: A hands-on, project-based introduction to programming</i> . No starch press, 2023.	
Recommended Texts	1- Liang, Y. Daniel. <i>Introduction to programming using Python</i> . Pearson, 2013. 2- McKinney, Wes. <i>Python for data analysis</i> . O'Reilly Media, Inc., 2022.	
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.

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information			
معلومات المادة الدراسية			
Module Title	Statistics and Probability		Module Delivery
Module Type			☒ Theory ☒ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☒ Seminar
Module Code	MU010602205		
ECTS Credits	4		
SWL (hr/sem)	100		
Module Level	1	Semester of Delivery	
Administering Department	Computer Science	College	College of Science
Module Leader	Muna Jaafar Raheem	e-mail	munmnmun@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Lecturer	Module Leader's Qualification	MSc.
Module Tutor	Hazim Michman Trao	e-mail	hazimmichman@uomustansiriyah.edu.iq
Peer Reviewer Name		e-mail	
Scientific Committee Approval Date	15/04/2024	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	This course will introduce the studies of the probability theory and statistics, with an emphasis on solving problems in computer science. Probability and statistics is an important foundation for computer science fields such as machine learning, artificial intelligence, computer graphics, randomized algorithms, image processing, and scientific simulations. Topics in probability include basic ideas in probability, Conditional Probability , Bayes' theorem, Independent Events ,discrete random variables, probability distributions, Special Discrete Random Variables , Bernoulli , Binomial and Poisson Random Variable. Topics in statistics include sample mean and variance, Estimating the Variance of a Normal Distribution, the sample correlation coefficient, and hypothesis testing.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	The Outcomes of this course Upon successful completion is that the students will have gained the necessary understanding on: 1. - laws of probability theory and its applications. 2. -Evaluate combination and permutations. 3. -use of Bayes' theorem. 4. - concept of Independent Events. 5. -concept of random variables. 6. - probability distributions and their properties. 7. -concept of sampling – concept of hypothesis testing. 8. -practical implementation of some of the probability and statistics concepts.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. 1. This course is an introductory text on a number of topics in statistics and probability, intended primarily for students undertaking a first degree in computer science. 2. The material of this course has been divided into three chapters covered on 15 weeks. 3. The basic areas covered: probability ,the sample statistics. 4. Each chapter begins with a clear statement of pertinent definitions, principles and theorems together with illustrative. 5. This is followed by sets of solved problems. The solved problems serve to illustrate and amplify the theory.

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	1. Active Learning: Encouraging students to actively engage with the material through activities such as problem-solving, group discussions, and hands-on projects. 2. Visualization: Using visual aids such as graphs, charts, and diagrams to illustrate statistical concepts and relationships 3. Real-World Examples: Providing real-world examples and applications of statistical concepts to help students understand how statistics is used in various fields such as science, business, and social science. 4. Technology Integration: Incorporating technology such as statistical software, simulation tools, and online resources into the learning process.. 5. Problem-Based Learning: Presenting students with real-world problems or case studies that require the application of statistical concepts to solve. 6. Assessment and Feedback: Providing regular opportunities for students to assess their understanding of the material through quizzes, assignments, and exams.. 7. Collaborative Learning: Encouraging collaboration among students through group projects, peer teaching, and collaborative problem-solving activities.
------------	--

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

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	20% (20)	3 and 10	LO #1, #2, #8
	Assignments	3	10% (10)	2, 7 and 11	LO #1, #2, #3, #4, #5, #6, #8

	Projects / Lab.				
	Report	1	10% (10)	13	All
Summative assessment	Midterm Exam	2hr	10% (10)	15	All
	Final Exam	4hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Basic Ideas in Probability, Sample spaces ,Events ,The Algebra of Events ,Axioms of Probability,
Week 2	Counting Outcomes ,Permutations
Week 3	Combinations , Quiz#1
Week 4	Conditional Probability ,Independent Events
Week 5	Discrete Random Variables
Week 6	Joint distributions ,Independent random variables ,Conditional distributions
Week 7	Expectation ,Variance and Standard Deviation ,Covariance
Week 8	Special Discrete Random Variables ,The Bernoulli Random Variable,
Week 9	The Binomial Random Variable ,The Poisson Random Variable
Week 10	Quiz#2,Sample Statistics, The Sample Mean, The Sample Variance
Week 11	Estimating the Variance of a Normal Distribution ,Samples from Finite Populations
Week 12	The Sample Correlation Coefficient
Week 13	Regression , Simple linear regression
Week 14	Maximum Likelihood Estimators ,Hypothesis Testing
Week 15	Midterm Exam #
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1	
Week 2	
Week 3	

Week 4	
Week 5	
Week 6	

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	Schaum's Outline Series " Probability and Statistics" Third Edition, Murray R. Spiegel, John J. Schiller, R. Alu Srinivasan, the McGraw-Hill Companies Inc,2009.	Yes
Recommended Texts	"Probability and Statistics for Computer Science" David Forsyth, Urbana, IL, USA , Copyright ,2018	No
Websites	https://doi.org/10.1007/978-3-319-64410-3	

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.				

MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information				
معلومات المادة الدراسية				
Module Title	The Crimes of the Baath Regime in Iraq		Module Delivery	
Module Type	Basic		☒ Theory ☐ Lecture ☐ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code	MU02			
ECTS Credits	2			
SWL (hr/sem)	50			
Module Level	2	Semester of Delivery		4
Administering Department	05	College	01	
Module Leader	Ahmed Akeel Abd		e-mail	ahmed.ageel@uomustansiriyah.edu.iq
Module Leader's Acad. Title	Assistant Lecturer		Module Leader's Qualification	M.Sc.
Module Tutor	No available		e-mail	E-mail
Peer Reviewer Name			e-mail	
Scientific Committee Approval Date	01/06/2023		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	تبصير هذه الاجيال باسلوب فكري وثائقي يكشف لهم عن جرائم هذا النظام البعثي بحق الشعب العراقي ، وتضمن هذا المنهج مفاتيح معرفية وضعت بيد الطالب الجامعي يقوى بها على مروجي اكاذيب البعث واعلامه المزيف
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1- تعريف الشباب الجامعي بجرائم ارتكبتها نظام جائر في العراق خاصة واثرها في المنطقة الاقليمية العامة و العالم كله 2- الوقوف على بطش هذا النظام بالوثائق والصور و التواريخ حتى يتسنى لهذا الجيل التأكد من كل تلك الجرائم بحق كل من انتهج سبيل الحق وحب الوطن و رفض النهج البعثي المجرم.
Indicative Contents المحتويات الإرشادية	تعليم الطالب الجامعي كيفية مواجهة الاعلام الزائف مستند على الوقائع التاريخية لما مر به العراق من حكم دكتاتوري وبالتالي مواجهة أي محاولة لاثارة الفكر البعثي او محاولة التمجيد له في الاوساط الشبابية .

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	الاستراتيجية الرئيسة التي سيتم تبنيها في تقديم هذه المادة ستركز على : المناقشة ، الاستجواب ، المحاضرة ، اوراق العمل ، السماح لطلبة بطرح الاسئلة.
-------------------	--

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	33	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	17	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	1
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	50		

Module Evaluation					
تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	4 and 9	LO #1, #2 and #7, #8
	Projects / Lab.	1	10% (10)	10	LO #5, #6 and #9
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	Midterm Exam	2hr	10% (10)	8	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
المنهاج الاسبوعي النظري	
	Material Covered
Week 1	مفهوم الجرائم واقسامها
Week 2	انواع الجرائم
Week 3	القرارات الصادرة من المحكمة
Week 4	الجرائم النفسية
Week 5	اليات الجرائم النفسية
Week 6	اثار الجرائم النفسية
Week 7	عسكرة المجتمع
Week 8	Midterm Exam
Week 9	موقف النظام البعثي من الدين
Week 10	صور انتهاكات حقوق الانسان
Week 11	اماكن السجون والاحتجاز لنظام البعث
Week 12	تدمير المدن والقرى سياسة الارض المحروقة
Week 13	احداث مقابر الابادة الجماعية المرتكبة من النظام
Week 14	جرائم المقابر الجماعية
Week 15	مقابر الابادة الجماعية لتي تعود للاحداث المذكورة
Week 16	Preparatory week before the final Exam

Learning and Teaching Resources		
مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	كتاب جرائم حزب البعث في العراق / وزارة التعليم العالي	نعم
Recommended Texts	المراجع ومواقع الانترنت	نعم
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.				