Academic Program Description Form

University name: Tikrit University
Faculty/ Institute: College of computer science and mathematics
Scientific Department: Computer Science
Academic or Professional program name: University Performance
Final certificate name: Bachelor`s
Academic System: Semester & Bologna system
Description preparation date: 1/9/2025
File completion date: 1/10/2025

Signature:
Head of department name:
Asst.Prof.Dr.Mohammed Aktham Ahmed

Date:

Signature:
Scientific associate name:
Asst.Prof.Dr. Majid Hamid Ali

Date:

الاستاذ المساعد الدكتور
ماجد حامد علي
معاون العميد للشؤون العلمية والدراسات العليا

The file is checked by:
Department of Quality Assurance and University Performance
Director of Quality Assurance and University Performance Department:
Lecturer: Yammr Abdulateef Hussein

Date:

Signature:

Approval of the dean

عميد كلية علوم الحاسوب والرياضيات

1. Program Vision

Providing distinguished academic programs in the field of computer science, both theoretical and applied, that comply with international standards of academic quality and meet the needs of the labor market. Encouraging and developing scientific research in the fields of computer science in general, and the fields of artificial intelligence, robotics, computational linguistics, and networks in particular.

2. Program Mission

The mission of the department is to prepare students professionally and academically with high-quality education through a model program equipped with all the teaching and learning tools the student needs and a highly qualified academic staff. The ultimate mission of the department is to provide distinguished education and a suitable academic environment in the fields of computer science.

3. Program Objectives

1. Providing modern and diverse educational programs in the fields of computer science, software and information technology, that meet international quality standards and keep pace with technical developments.
2. Graduating distinguished students with technical and creative skills and the ability to solve complex problems in the fields of computer science and software.
3. Providing an integrated educational environment based on the latest tools and technologies to improve
4. Supporting students and faculty members in adopting a culture of innovation and technological entrepreneurship, and encouraging the creation of innovative technical projects.
5. Providing distinguished graduate programs in software and computer science to qualify advanced academic and research competencies.
6. Instilling the values of integrity, credibility and social responsibility in all aspects of education and scientific research.
7. Providing training programs and workshops to raise the efficiency of students, graduates and faculty members and enhance opportunities for continuous learning.

4. Programmatic accreditation:
1. The student learns programming languages. 2. The ability to find scientific software solutions to community problems. 3. Develop the student's skills in building smart systems that depend on the basis of analysis and inference. 4. Provide the student with the basic rules in building software systems based on the basics of software engineering. 5. Increase the student's information on the basics of implementing software systems by understanding the mechanism of computer operation.

5. Other external influences:
1. Through the scientific conference for students. 2. The quarterly scientific seminar of the department. 3. Discussion sessions for professors. 4. Research seminars.

Notes	Unit of study	Number of courses	Program Structure
	12	6	Institutional Requirements
	18	6	Faculty Requirements
	84	20	Department Requirements
	Completed	1	Summer Training
			Other

* Notes can include whether the course is core or optional.

7. Program Description				
Credit Hours		Course Name	Course Code	Level
Lab	Theoretical			
2	2	Programming Fundamentals	TU010101107	Stage 1/Semester 1
2	2	Principles of Computer Structure	TU010101109	Stage 1/Semester 1
	4	Discrete Structures 1	TU010101108	Stage 1/Semester 1
	4	Differential and Integral Calculus 1	TU010101110	Stage 1/Semester 1
	2	English Language 1	TU010101101	Stage 1/Semester 1
	2	Democracy and Human Rights	TU010101102	Stage 1/Semester 1
-	4	Computer Architecture	TU010102002	Stage 2/ First Semester
2	2	Network protocols	TU010102015	Stage 2/ First Semester
2	2	Data Structures 1	TU010102005	Stage 2/ First Semester
2	2	Database basics	TU010102016	Stage 2/ First Semester
2	2	Entity Programming	TU010102007	Stage 2/ First Semester
-	2	Ba'ath Party Crimes in Iraq	TU010102008	Stage 2/ First Semester
2	2	Switching, routing, and wireless basics	TU010103014	Stage 3/ First Semester
2	2	Website programming	TU010103015	Stage 3/ First Semester
2	2	Mobile application development	TU010103016	Stage 3/ First Semester
2	2	Operating system	TU010103004	Stage 3/ First Semester
2	2	Data science	TU010103017	Stage 3/ First Semester
2	2	Software Engineering	TU010103006	Stage 3/ First Semester
2	2	Digital Image Processing	TU010104001	Stage 4/ First Semester
2	2	Operating Systems	TU010104002	Stage 4/ First Semester
2	2	Computer Security	TU010104003	Stage 4/ First Semester
-	4	Computer Networks	TU010104004	Stage 4/ First Semester
2	2	Machine learning	TU010104005	Stage 4/ First Semester

2	**2**	Graduation Project	TU010104012	Stage 4/ First Semester

7. Program Description					
Credit Hours			Course Name	Course Code	Level
Tut.	Lab	Theoretical			
2	2	2	Advanced Programming	TU010101113	Stage 1/Second Semester
	2	2	Algorithm design	TU010101111	Stage 1/Second Semester
		4	Theory of computation	TU010101114	Stage 1/Second Semester
	2	2	Logic Design	TU010101112	Stage 1/Second Semester
	2	2	Web Programming 1	TU010101115	Stage 1/Second Semester
		2	Arabic Language	UOT001	Stage 1/Second Semester
	2	2	Artificial intelligence	TU010101223	Stage 2/ Second Semester
	2	2	Relational databases and structured query language	TU010101225	Stage 2/ Second Semester
	-	2	English Language II	UOT021	Stage 2/ Second Semester
2	2	2	Visual programming	TU010101224	Stage 2/ Second Semester
	2	2	Statistics and probability	TU010101222	Stage 2/ Second Semester
		2	Arabic Language2	UOT011	Stage 2/ Second Semester
	2	2	Machine Learning	TU010101320	Stage 3/ Second Semester
	2	2	Compilers	TU010101321	Stage 3/ Second Semester
	2	2	Web Development Frameworks	TU010101322	Stage 3/ Second Semester
		2	Research Methodology	TU010101323	Stage 3/ Second Semester

	2	**2**	Internet of Things	TU010101324	Stage 3/ Second Semester
	2	**2**	Operating systems 2	TU010101325	Stage 3/ Second Semester
		4	Distributed systems	433ح	Stage 4/ Second Semester
	2	**2**	Computer Networks	342ح	Stage 4/ Second Semester
	2	**2**	Multimedia and Network Security	463ح	Stage 4/ Second Semester
	2	**2**	Data mining	432ح	Stage 4/ Second Semester
	2	**2**	Neural Networks	412ح	Stage 4/ Second Semester
	2	**2**	Graduation Project	492ح	Stage 4/ Second Semester

8. Expected Learning Outcomes of the Program

Knowledge

1. Enable students to gain knowledge and understanding of computer basics
2. Enable students to gain knowledge and understanding of computer applications
3. Enable students to gain knowledge and understanding of computer programming
4. Enable students to gain knowledge and understanding of computer networks
5. Enable students to gain knowledge and understanding of multimedia

Enable students to gain knowledge and understanding of data science and mining.

Skills

Memorization and analysis skills	Scientific skills
• **Empowering students to obtain knowledge and understanding of computer programming** • **The ability to remember information accurately and in an organized manner.** • **The ability to retrieve information quickly and effectively when needed.**	• **Proficiency in different programming languages such as Java, C, C++, etc.** • **Understanding the basic processes and concepts of software development in an effective and organized manner.** • **Understanding how to build and manage information networks.**

• **The ability to prioritize activities and allocate time effectively.**	• **Understanding security threats and how to protect data and networks from them.** • **Learn artificial intelligence and machine learning and understand the models and algorithms used in developing intelligent systems.** • **The ability to extract and analyze data effectively to obtain valuable insights.** • **Learn the basics of saving and viewing data using different databases.** • **Web application development**
Values	
Learning Outcome Statement 4	Enable students to gain knowledge and understanding of computer networks
Learning Outcome Statement 5	Enable students to gain knowledge and understanding of multimedia
9. Teaching and learning strategies:	
1- Providing students with the basics and additional topics related to the previous educational outcomes of skills to solve scientific problems. 2- Solving a set of scientific examples by the academic staff. 3- Asking students during the lecture to solve some scientific problems 4. Visiting all scientific computer science laboratories by the academic staff.	
10. Evaluation methods:	
1- Standard Quality Systems (standards Quality) 2- Daily exams with self-solved homework questions. 3- Points for sharing questions for a competition related to the subject matter. 4- Points for specific homework assignments.	

11. The teaching staff

The teaching staff members

ت	الاسم	الشهادة	اللقب العلمي	التخصص		اعداد الهيئة التدريسية	
				عام	خاص	ملاك	محاضر
.1	محمود ماهر صالح علاوي الجبوري	دكتوراه	استاذ مساعد	علوم الحاسوب	ذكاء اصطناعي و تطبيقاته	ملاك	-
.2	ماجد حامد علي جاسم	دكتوراه	استاذ مساعد	علوم حاسبات	تكنولوجيا المعلومات وتطبيقاتها	ملاك	-
.3	محمد اكثم احمد الطه	دكتوراه	استاذ مساعد	علوم حاسبات	ذكاء اصطناعي و تطبيقاته	ملاك	-
.4	مشاري عايد عسكر ملاش الشمري	دكتوراه	أستاذ	علوم الحاسوب	تكنولوجيا معالجة الإشارة الرقمية وتطبيقاتها	ملاك	-
.5	ميثم مصطفى حمود جزاع المرعاوي	دكتوراه	استاذ	علوم حاسوب	امنية البيانات	ملاك	-
.6	سعدي حمد ثلج احمد	دكتوراه	استاذ مساعد	علوم حاسبات	تكنلوجيا قواعد البيانات وتطبيقاتها	ملاك	-
.7	فراس ليث خليل محمود	دكتوراه	استاذ مساعد	علوم حاسبات	تكنولوجيا المعلومات	ملاك	-
.8	احمد محمد شريف عبد الله العباسي	دكتوراه	مدرس	اللغة العربية	الأدب العربي	ملاك	-
.9	كمال مجيد وهاب	دكتوراه	مدرس	ادارة اعمال	الإدارة والتسويق	ملاك	-
.10	همام نعمة حسين درويش الجبوري	دكتوراه	مدرس	هندسة كهرباء وحاسبات	شبكات الاتصالات	ملاك	-
.11	محمود حسين علوان فرج العبيدي	دكتوراه	مدرس	ادارة اعمال	تسويق	ملاك	-
.12	علي خلف مصطفى	دكتوراه	مدرس	ادارة اعمال	ادارة اعمال	ملاك	-
.13	حارث عبدالله حسين عبدالله الجبوري	ماجستير	أستاذ	علوم الحاسوب	تكنلوجيا الويب وتطبيقاتها	ملاك	-
.14	ارمانيسة نعمان حسون علي الكراعي	ماجستير	استاذ مساعد	علوم حاسوب	ذكاء اصطناعي	ملاك	-
.15	نايف محمد حماش خلف	ماجستير	استاذ مساعد	علوم الحاسوب	تكنولوجيا المعلومات وتطبيقاتها	ملاك	-
.16	سلوى خالد عبد اللطيف السالم	ماجستير	استاذ مساعد	علوم حاسبات	ذكاء اصطناعي	ملاك	-
.17	محمد وليد نوفان عسكر التكريتي	ماجستير	استاذ مساعد	علوم حاسوب	معالجة صور	ملاك	-
.18	وسام داود عبد الله خلف الجبوري	ماجستير	استاذ مساعد	علوم الحاسوب	شبكات وانترنت الاشياء وتطبيقاتها	ملاك	-
.19	عمار فاروق عباس وهيب التكريتي	ماجستير	مدرس	هندسة حاسبات	الذكاء الاصطناعي وتطبيقاته	ملاك	-
.20	حذيفة محمد كنوش حمد الجبوري	ماجستير	مدرس	علوم الحاسوب	الذكاء الاصطناعي وتطبيقاته	ملاك	-
.21	مهند داود سلمان رومي الرومي	ماجستير	مدرس	علوم الحاسوب	الذكاء الاصطناعي وتطبيقاته	ملاك	-
.22	اسراء رافع عبد القادر محمد الحديثي	ماجستير	مدرس	علوم حاسوب	ذكاء اصطناعي	ملاك	-
.23	يحيى ليث خليل محمود الأمين	ماجستير	مدرس	علوم حاسوب	الذكاء الاصطناعي وتطبيقاته	ملاك	-
.24	ريم ضياء اسماعيل هلال التكريتي	ماجستير	مدرس	علوم حاسوب	الذكاء الاصطناعي	ملاك	-
.25	مصطفى لطيف فاضل لطيف جميلي	ماجستير	مدرس مساعد	علوم الحاسوب	علوم الحاسوب	ملاك	-
.26	كهلان فائق حسن احمد	ماجستير	مدرس مساعد	علوم حاسوب	تقنيات الشبكات والاتصالات اللاسلكية	ملاك	-
.27	لؤي ابراهيم خلف عبد الله	ماجستير	مدرس مساعد	علوم حاسبات	نظم معلومات	ملاك	-
.28	عقبه سالم عطيه احمد الجبوري	ماجستير	مدرس	علوم حاسوب	الذكاء الاصطناعي	ملاك	-

11. The teaching staff

The teaching staff members

ت	الاسم	الشهادة	اللقب العلمي	التخصص		اعداد الهيئة التدريسية	
				عام	خاص	ملاك	محاضر
.29	حسن ناجي علي سبع الشمري	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.30	قيس عامر حميد محمد الناصري	ماجستير	مدرس	علوم حاسوب	الذكاء الاصطناعي	ملاك	-
.31	احمد فايق صابر عجوان المرسومي	ماجستير	مدرس	اللغة الانكليزية	طرائق تدريس	ملاك	-
.32	نبأ احمد سعيد ابراهيم	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.33	اسراء أكرم بشير محمود	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.34	عمر عادل لبيب عبد اللطيف	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.35	مهند حاتم رمضان قدوري	ماجستير	مدرس مساعد	علوم الحاسوب	علوم حاسبات	ملاك	-
.36	مروه حسن حديد خلف الجبوري	ماجستير	مدرس مساعد	علوم الحاسوب	علوم حاسبات	ملاك	-
.37	اسراء محمد احمد حسين البياتي	ماجستير	مدرس مساعد	علوم الحاسوب	علوم حاسبات	ملاك	-
.38	عدي وسمي عبود محمد	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.39	سعاد محمد شكور	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.40	رشا عبد الله يوسف	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.41	معاذ فرمان اسماعيل	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.42	نور وليد خالد	ماجستير	مدرس مساعد	علوم حاسوب	علوم حاسبات	ملاك	-
.43	محمد حميد عيسى	ماجستير	مدرس مساعد	ادارة اعمال	ادارة اعمال	ملاك	-

Professional Development

Orientation of New The teaching staff

For visiting faculty, a specific process can be designed to guide and support them during their time at the college. The following is a brief description of the process used to guide visiting faculty:

1- Welcome and Orientation: Visiting faculty are warmly welcomed and provided with an orientation about the college's culture, policies, and resources. They receive an introduction to the department they will be working with and are introduced to key staff and support personnel.

2- Integration: Visiting faculty are integrated into their department during their visit. They meet with department chairs, fellows, and administrative staff who brief them on departmental expectations, teaching assignments, research opportunities, and

departmental initiatives.

3- Evaluation and Feedback: The department conducts periodic evaluations of faculty performance, providing constructive feedback to support their professional development. This evaluation process may include classroom observations, evaluation of research outputs, and discussions with the department chair and department chair.

4- Continuous Professional Development: Professional development ensures that visiting faculty members have continuous professional development opportunities. They are introduced to relevant workshops, training programs and conferences that match their areas of interest and expertise.

Professional development for teaching staff members

The Professional Development Department plays an important role in facilitating the academic and professional development of faculty members within the college or department. The following is a brief description of the plan and arrangements for faculty development in areas such as teaching and learning strategies, assessment of learning outcomes, and professional development:

Teaching and Learning Strategies:

1- The department organizes workshops, seminars, and training courses that focus on effective teaching and learning strategies.

2- The department encourages the use of technology in the classroom and provides support for integrating educational technologies into teaching practices.

Assessment of Learning Outcomes:

1- The department provides guidance and resources on designing assessments that align with learning outcomes and program objectives.

2- Faculty members receive training on various assessment methods.

Professional Development:

1- The department facilitates continuing professional development opportunities for faculty members.

2- The department organizes conferences, seminars, and guest lectures featuring

lecturers in various academic disciplines.

3- Faculty members are encouraged to attend external professional development events.

Mentoring and Peer Collaboration:

1- The department promotes a culture of mentoring and collaboration among faculty members.

2- The department facilitates mentoring programs where experienced faculty members guide and support their colleagues, especially new faculty members.

12. Acceptance criteria.

For morning study:
Students are accepted through the central admission system and according to the instructions of the Iraqi Ministry of Higher Education and Scientific Research.

For evening study:
Students are accepted by applying to the college by submitting some required documents, and students are accepted based on the instructions and decisions of the Iraqi Ministry of Higher Education and Scientific Research.

13. The most important sources of information about the program

1- Methodological and support books
2- Lectures of the instructor
3- Internet for the purpose of obtaining research.

14. Program Development Plan:

Academic program development in colleges involves a systematic and collaborative approach. The following outlines an appropriate program development plan:

Needs Assessment:
1- Conduct a comprehensive needs assessment to identify gaps and opportunities in the academic landscape.
2- Analyze market trends, recruitment requirements, and industry requirements to align the program with current and future needs.

Faculty Development:
1- Provide faculty members with professional development opportunities, including workshops, conferences, and training programs.
2- Support faculty members in keeping up with developments in the field through research and scholarly activities.

Resource and Infrastructure Allocation:

1- Identify resource needs to implement the program effectively.
2- Allocate sufficient financial, technological, and human resources to support the program.
3- Ensure that the program has access to the necessary infrastructure, including classrooms, laboratories, equipment, and specialized facilities.

Cooperation and Partnerships:
1- Establish partnerships for internships, cooperative education programs, and research projects.
2- Leverage external expertise and resources to enhance the quality and relevance of the program.

Monitoring and Evaluation:
1- Develop a comprehensive evaluation plan to assess student learning outcomes and program effectiveness.
2- Use evaluation data to identify areas for improvement and make data-informed decisions to enhance the program.

Values				Skills				Knowledge				Course Name	Course Code	Year /Level
ج4	ج3	ج2	ج1	ب4	ب3	ب2	ب1	أ4	أ3	أ2	أ1			
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Programming Fundamentals	TU010101107	First stage First semester 2026-2025
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Principles of Computer Structure	TU010101109	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Discrete Structures 1	TU010101108	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Differential and Integral Calculus 1	TU010101110	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	English Language 1	TU010101101	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Democracy and Human Rights	TU010101102	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Computer Architecture	TU010102002	second stage First semester
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Network protocols	TU010102015	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Data	TU010102005	

												Structures 1		2026-2025
✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	Database basics	TU010102016	
✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	Entity Programming	TU010102007	
✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	Ba'ath Party Crimes in Iraq	TU010102008	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Switching, routing, and wireless basics	TU010103014	Third stage First semester 2026-2025
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Website programming	TU010103015	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Mobile application development	TU010103016	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Operating system1	TU010103004	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Data science	TU010103017	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Software Engineering	TU010103006	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Digital Image Processing	TU010104001	Fourth stage First semester 2026-2025
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Operating Systems	TU010104002	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Computer Security	TU010104003	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Computer Networks	TU010104004	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Machine learning	TU010104005	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Graduation Project	TU010104012	

*Please tick the boxes corresponding to the individual learning outcomes of the programmer being assessed

Values				Skills				Knowledge				Course Name	Course Code	Year /Level
4ج	3ج	2ج	1ج	4ب	3ب	2ب	1ب	4أ	3أ	2أ	1أ			
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Advanced Programming	TU010101113	First stage Second Semester -2025 2026
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Algorithm design	TU010101111	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Theory of computation	TU010101114	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Logic Design	TU010101112	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Web Programming 1	TU010101115	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Arabic Language	UOT001	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Artificial intelligence	TU010101223	second stage Second Semester -2025 2026
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Relational databases and structured query language	TU010101225	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	English Language II	UOT021	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Visual programming	TU010101224	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Statistics and probability	TU010101222	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Arabic Language 2	UOT011	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	Machine Learning	TU010103014	Third stage Second Semester -2025 2026
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Compilers	TU010101320	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Web Development Frameworks	TU010101321	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Research Methodology	TU010101322	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Internet of Things	TU010101323	

✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Operating systems 2	TU010101324	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Distributed systems	433ح	Fourth stage Second Semester -2025 2026
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Computer Networks	342ح	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Multimedia and Network Security	463ح	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Data mining	432ح	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Neural Networks	412ح	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Graduation Project	492ح	

MODULE DESCRIPTION FORM
نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	Programming Fundamentals			**Module Delivery**	
Module Type	Core			☐ **Theory** ☒ **Lecture** ☒ **Tutorial** ☒ **Lab** ☐ **Practical** ☐ **Seminar**	
Module Code	TU010101107				
ECTS Credits	8				
SWL (hr/sem)	200				
Module Level		**1**	**Semester of Delivery**		**1st**
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Mohanad Hatem Ramadhan		**e-mail**	Mohanad.H.Ramadhan@tu.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		Master
Module Tutor	Maath Farman Ismaeel		**e-mail**	Maath.F.Ismaeel@tu.edu.iq	
Peer Reviewer Name		Mahammed Aktham	**e-mail**		
Scientific Committee Approval Date		1/9/2025	**Version Number**	2.0	

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

Module Aims, Learning Outcomes and Indicative Contents
أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Aims أهداف المادة الدراسية	1. To introduce students to the fundamental principles and concepts of programming. 2. To familiarize students with the syntax and structure of the C++ programming language. 3. To develop students' problem-solving skills and algorithmic thinking. 4. To enable students to design, implement, and test programs using C++ to solve computational problems. 5. To provide students with hands-on experience in programming through practical exercises, assignments, and projects. 6. To promote the use of modular programming techniques for creating reusable and maintainable code. 7. To enhance students' ability to debug and troubleshoot programs effectively. 8. To develop students' communication skills in expressing programming concepts and solutions clearly and effectively. 9. To prepare students for advanced programming courses and real-world software development scenarios.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Demonstrate a solid understanding of the fundamental principles and concepts of programming. 2. Utilize the syntax and structure of the C++ programming language to write well-structured and efficient code. 3. Apply problem-solving skills and algorithmic thinking to develop solutions for a variety of computational problems. 4. Design, implement, and test programs using C++ to solve specific tasks and challenges. 5. Utilize modular programming techniques to create reusable and maintainable code. 6. Debug and troubleshoot programs effectively using appropriate debugging techniques and tools. 7. Collaborate and work effectively in teams to complete programming projects. 8. Communicate programming concepts, solutions, and ideas clearly and effectively, both orally and in written form. 9. Demonstrate a readiness to progress to more advanced programming courses or pursue a career in software development.
Indicative Contents المحتويات الإرشادية	1. Introduction to Computer Science: - Overview of computer science as a discipline

- Key concepts and principles in computer science
- Role of programming in computer science

2. Introduction to Computers, Binary System, and Information Representation:
- Basics of computer architecture and components
- Understanding the binary system and its significance in computing
- Conversion between binary and decimal.
- Representation of different data types in computers
- ASCII and Unicode for character encoding

3. Algorithm Design and Problem Solving:
- Understanding algorithms and problem-solving strategies
- Analyzing problem requirements and designing algorithmic solutions
- Time and space complexity analysis
- Representing algorithms with Pseudocode and Flowcharts:
  - Using pseudocode as a high-level representation of algorithms
  - Writing pseudocode to describe the logic and steps of an algorithm
  - Understanding flowcharts as visual representations of algorithms
  - Basic flowchart symbols and their meanings
  - Creating flowcharts to represent the flow of control in algorithms

4. Introduction to C++:
- History and features of the C++ programming language
- Setting up a C++ development environment
- Basic syntax and structure of C++ programs

5. Variables and Data Types:
- Declaring and initializing variables
- Fundamental data types (integers, floating-point numbers, characters)
- Working with constants and literals

6. Operators and Expressions:
- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators

7. Control Structures:
- Decision-making with if-else statements
- Switch statements for multiple choices
- Repetition with loops (while, do-while, for)
- Handling user input and validation

8. Functions:

	- Function declaration and definition - Parameters and argument passing - Return values and function overloading - Scope and lifetime of variables

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain programming concepts, C++ syntax, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively.

	4. Programming Assignments: Assignments will be given to students to reinforce their understanding of programming concepts and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using C++. 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	60	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	140	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	9.4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Due Week	Relevant Learning Outcome
Formative assessment	**Quizzes**	3	15% (15)	5, 7, 11	
	In-Class Activity	1	5% (5)	9	
	Assignments	2	10% (10)	6, 12	
	Projects	1	10% (10)	5-14	
	Report				
Summative assessment	**Midterm Exam**	1 hr.	10% (10)	11	
	Final Exam	2 hr.	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction to Computer Science, Computers Components, Binary and Info Representation
Week 2	Algorithms Design and Writing pseudocode
Week 3	Algorithms design and Drawing Flowchart
Week 4	Introduction to Programming Languages (History, Categories, Main Differences) and, Introduction to Programming in C++ (Program Structure and Coding Environment)
Week 5	Variables, Datatypes, Output, and Input
Week 6	Operations (Arithmetic and Assignment) and Math Functions
Week 7	Operations (Comparison and Logical)
Week 8	Flow Control (if – else)
Week 9	Flow Control (switch – case)
Week 10	Loops (counter and cumulative variables)
Week 11	Uncountable Loops
Week 12	Nested Loops
Week 13	Functions
Week 14	building a TikTacToe Game
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Using Operating System, Creating Files and Folders, Writing Text)
Week 2	Difference among (Text Editor, Word Processor, Code Editor and IDE)
Week 3	Drawing (Darg and drop) Flowcharts
Week 4	**Installing C++ coding environment and running Hello World program**
Week 5	Running Examples on Variables, Datatypes, Output, and Input
Week 6	Running Examples on Operations (Arithmetic and Assignment) and Math Functions
Week 7	Running Examples on Operations (Comparison and Logical)
Week 8	Running Examples on Flow Control (if – else)
Week 9	Running Examples on Flow Control (switch – case)
Week 10	Running Examples on Loops (counter and cumulative variables)
Week 11	Running Examples on Uncountable Loops

Week 12	Running Examples on Nested Loops
Week 13	Running Examples on Functions
Week 14	Fixing problems in students' projects
Week 15	Applying instructor's feedback on students' projects

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Stroustrup, Bjarne - Programming_ principles and practice using C++-Addison-Wesley (2015)	Yes
Recommended Texts	Olsson, Mikael - C++20 Quick syntax reference: a pocket guide to the language, apis, and library	No
Websites	https://learncpp.com/ https://harmash.com/tutorials/cplusplus/overview	

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	Principles of Computer Organization			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	TU010101109				
ECTS Credits	6				
SWL (hr/sem)	150				
Module Level	1		**Semester of Delivery**	1st	
Administering Department	Computer science		**College**	CCSM	
Module Leader	Nayif Mohamed		**e-mail**		
Module Leader's Acad. Title	Asst. Prof.		**Module Leader's Qualification**	Master	
Module Tutor	Ammar Farooq Abbas		**e-mail**	ammar.abbas@tu.edu.iq	
Peer Reviewer Name	Mohamed Aktham		**e-mail**		
Scientific Committee Approval Date	1/09/2025		**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. Understanding the Basic Components: The module aims to familiarize students with the fundamental components of computer systems, including the CPU (Central Processing Unit), memory, input/output devices, and storage devices. Students should gain an understanding of the purpose and functionality of each component. 2. Learning about Instruction Set Architecture: Students will learn about different instruction set architectures, such as the von Neumann architecture, Harvard architecture, and reduced instruction set computing (RISC) architecture. The

	module aims to provide an understanding of the design principles behind instruction sets and their impact on computer performance. 3. Understanding Processor Design: The module aims to delve into the design and operation of processors, including topics such as control unit design, arithmetic logic units (ALUs), pipelining, and parallel processing. Students should gain an understanding of how processors execute instructions and the factors affecting their performance. 4. Studying Memory Systems: This module aims to cover different types of memory systems, including cache memory, main memory (RAM), and secondary storage (hard drives, solid-state drives). Students should understand the organization, hierarchy, and access methods of these memory systems, as well as their impact on system performance. 5. Exploring Input/Output Systems: The module aims to provide an understanding of input/output (I/O) systems and their interface with the CPU and memory. Topics covered may include I/O devices, interrupt handling, I/O techniques (e.g., programmed I/O, interrupt-driven I/O, DMA), and I/O performance considerations. 6. Discussing Parallel Processing and Multiprocessing: Students may learn about the concepts of parallel processing and multiprocessing, including techniques such as parallel algorithms, multi-core processors, and parallel architectures. The module aims to explore the benefits and challenges of parallel computing. 7. Analyzing Performance Evaluation: The module may introduce techniques for evaluating the performance of computer systems, including benchmarking, measurement metrics, and performance analysis. Students should gain the ability to assess and compare the performance of different computer systems. 8. Understanding Operating Systems: The module aims to provide students with an understanding of operating systems, including their purpose, functionalities, and components. Topics covered may include process management, memory management, file systems, device management, and scheduling algorithms. Students should gain knowledge of the key concepts and mechanisms used in operating systems. 9. Discussing System Maintenance and Administration: The module may cover system maintenance and administration principles, including software updates, patch management, system monitoring, troubleshooting, and system backup and recovery. Students should gain knowledge of the practices and tools used in maintaining and administering system software. 10. Understanding Software Functionality: The module aims to familiarize students with the functionality and features of application software, specifically focusing on tools and applications within productivity suites like Microsoft Office. This includes word processing, spreadsheet analysis, presentation creation, and database management.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand the basic components of computer systems, including the CPU, memory, input/output devices, and storage devices. 2. Demonstrate knowledge of different system architectures and their impact on computer performance. 3. Understand instruction set architectures and their design principles. 4. Explain the design and operation of processors, including control unit design,

ALUs, pipelining, and parallel processing.

5. Understand memory systems, including cache memory, main memory, and secondary storage, and their impact on system performance.
6. Describe input/output systems and their interface with the CPU and memory, including I/O devices, interrupt handling, and I/O techniques.
7. Discuss parallel processing and multiprocessing concepts, including parallel algorithms and architectures.
8. Analyze the performance of computer systems using benchmarking and performance evaluation techniques.
9. Understand the purpose, functionalities, and components of operating systems.
10. Demonstrate knowledge of system programming, including system calls, libraries, drivers, and low-level programming.
11. Understand virtualization and emulation principles and their applications.
12. Demonstrate knowledge of system security concepts and practices.
13. Understand system maintenance and administration principles, including software updates, system monitoring, and troubleshooting.
14. Demonstrate proficiency in using application software tools and features within the productivity suite (e.g., Microsoft Word, Excel, PowerPoint):
    a. Create and edit documents, spreadsheets, presentations, and databases effectively and efficiently.
    b. Utilize formatting options, templates, and styles to enhance the visual appeal and professionalism of documents.
    c. Use formulas, functions, and data analysis tools to manipulate and analyze data in spreadsheets.
    d. Design visually appealing and engaging presentations using multimedia elements and slide transition effects.
15. Apply best practices for efficient software usage:
    a. Navigate the user interface and menu options to locate and utilize software features effectively.
    b. Utilize keyboard shortcuts and automation features to enhance productivity and streamline tasks.
    c. Optimize workflow and utilize time-saving techniques within the software.
16. Communicate effectively through the creation of professional-looking documents and presentations:
    a. Create clear, concise, and well-organized documents with appropriate formatting, headings, and sections.
    b. Design visually appealing slides with effective use of text, images, charts, and other multimedia elements.
    c. Utilize collaboration and sharing features to facilitate teamwork and effective communication.
17. Analyze and visualize data using spreadsheet software:
    a. Organize and manipulate data effectively using tables, filters, and sorting features.
    b. Utilize formulas, functions, and data analysis tools to perform calculations and derive insights from data.
    c. Create informative charts, graphs, and pivot tables to visualize data and present findings effectively.
18. Demonstrate problem-solving skills and troubleshoot common software issues:

	d. Identify and resolve common software-related challenges and errors. a. Seek out appropriate resources and support channels to resolve problems independently. b. Apply critical thinking and problem-solving strategies when encountering software-related issues. 19. Manage information effectively and maintain data integrity and security: a. Organize and store files and data in a structured and easily retrievable manner. b. Manage document versions, track changes, and utilize collaboration features to ensure data integrity. c. Apply security measures to protect sensitive information and maintain confidentiality. 20. Demonstrate an ability and willingness to learn and adapt to new software features and advancements: a. Stay updated with software updates and new features within the application software. b. Explore online resources, tutorials, and communities to expand knowledge and skills. c. Display a mindset of continuous learning and adaptability in using application software.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following: 1. Basic Components of Computer Systems: [6 hours] • CPU (Central Processing Unit) • Memory (Primary and Secondary) • Input/Output Devices • Storage Devices 2. System Architecture: [6 hours] • Instruction Sets • Data Representation • Memory Hierarchy • Caches and Cache Organization 3. Processor Design: [6 hours] • Control Unit Design • Arithmetic Logic Units (ALUs) • Pipelining and its Challenges • Parallel Processing and Multi-core Processors 4. Memory Systems: [6 hours] • Cache Memory • Main Memory (RAM) • Secondary Storage (Hard Drives, Solid-State Drives) • Virtual Memory 5. Input/Output Systems: [6 hours] • I/O Devices

- Interrupt Handling
- Programmed I/O, Interrupt-Driven I/O, DMA
- I/O Performance Considerations

6. Parallel Processing and Multiprocessing: [1.5 hours]
   - Parallel Algorithms
   - Multi-core Processors
   - Parallel Architectures
   - Performance Considerations and Challenges
7. Performance Evaluation: [1.5 hours]
   - Benchmarking
   - Measurement Metrics
   - Performance Analysis Techniques
   - Comparing and Assessing Computer System Performance

Indicative Contents for System Software:

1. Operating Systems: [6 hours]
   - Purpose and Functions of Operating Systems
   - Process Management
   - Memory Management
   - File Systems
   - Device Management
   - Scheduling Algorithms
2. System Maintenance and Administration: [3 hours]
   - Software Updates and Patch Management
   - System Monitoring and Troubleshooting
   - System Backup and Recovery
   - Performance Tuning
3. System Maintenance and Administration: [3 hours]
   - Software Updates and Patch Management
   - System Monitoring and Troubleshooting
   - System Backup and Recovery
   - Performance Tuning

Indicative Contents for Studying Application Software (e.g., Microsoft Office):

1. Introduction to Productivity Suites: [3 hours]
   - Overview of productivity suites and their significance in various professional contexts.
   - Introduction to the features and components of popular productivity suites like Microsoft Office.
2. Word Processing Software (e.g., Microsoft Word): [3 hours]
   - Creating, editing, and formatting documents.

	- Working with text, paragraphs, and styles. - Inserting and formatting images, tables, and other graphical elements. - Utilizing document templates and mail merge functionality. - Collaboration and reviewing features. 3. Spreadsheet Software (e.g., Microsoft Excel): [3 hours] - Creating, formatting, and managing spreadsheets. - Working with formulas, functions, and calculations. - Data manipulation and analysis techniques. - Creating charts, graphs, and pivot tables. - Spreadsheet automation and advanced features. 4. Presentation Software (e.g., Microsoft PowerPoint): [3 hours] - Creating and formatting slides. - Inserting and formatting text, images, shapes, and multimedia elements. - Utilizing slide layouts, transitions, and animations. - Designing effective presentations for different audiences and purposes. - Collaboration and presentation delivery techniques. 5. Document Security and Protection: [3 hours] - Applying document security measures, password protection, and encryption. - Managing document permissions and access controls. - Ensuring data integrity and confidentiality within productivity suites.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Active Learning: • Actively engage with the material by reading, taking notes, and asking questions. • Participate in discussions, group activities, and practical exercises. • Seek opportunities for hands-on practice and application of concepts through exercises, projects, and real-world examples. 2. Practical Application: • Apply the learned concepts in practical scenarios, such as building and analyzing computer systems, designing software solutions, or creating documents and presentations using application software. • Seek out real-world examples and case studies to understand how the concepts are applied in practice. • Undertake practical projects or assignments to gain hands-on experience. 3. Experimentation and Exploration: • Explore and experiment with different computer architectures, operating systems, and application software to gain a deeper understanding.

- Set up virtual environments or use online simulators to experiment with various software configurations and settings.
- Explore additional features and functionalities beyond the basics, going beyond the prescribed curriculum to expand knowledge.

4. Collaborative Learning:
    - Engage in group discussions, study groups, or online forums to share knowledge, exchange ideas, and discuss complex topics.
    - Collaborate with peers on projects or assignments to enhance learning through teamwork and shared problem-solving.
5. Continuous Practice and Review:
    - Regularly practice the learned concepts through exercises, assignments, or self-assessment quizzes.
    - Review and reinforce knowledge through periodic revision of key topics.
    - Seek feedback from instructors or mentors to identify areas for improvement and further learning.
6. Utilize Resources:
    - Make use of textbooks, lecture notes, online tutorials, and supplementary resources to enhance understanding and clarify concepts.
    - Explore online resources, forums, and communities related to the subject to access additional learning materials, discussion threads, and expert guidance.
7. Stay Updated:
    - Keep up-to-date with the latest advancements, trends, and updates in the field of computer organization, system software, and application software.
    - Follow relevant industry news, blogs, and publications to stay informed about emerging technologies and best practices.
8. Time Management and Regular Study:
    - Allocate dedicated study time for each subject to ensure consistent progress and understanding.
    - Break down the learning material into manageable chunks and create a study schedule.
    - Avoid procrastination and maintain discipline in adhering to the study plan.
9. Seek Support:
    - Seek guidance and support from instructors, teaching assistants, or mentors when encountering difficulties or complex concepts.
    - Participate in office hours or seek clarification during lectures or tutorials.

	10. Reflect and Apply Learning: • Regularly reflect on the learning process, strengths, and areas for improvement. • Apply the acquired knowledge in practical situations to reinforce understanding and develop practical skills. • Connect the learned concepts to real-world applications and scenarios

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	92	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6.13
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	58	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	3.86
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)	5, 10	LO #1-5 , LO# 10-14
	Assignments	2	10% (10)	2, 12	LO #10-14, LO#14-16
	Seminar	1	10% (10)	11	LO # 5-11
	Report	1	10% (10)	12	LO # 5-11
Summative assessment	**Midterm Exam**	2 hr	10% (10)	7	LO # 1-17
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction: Computer Overview
Week 2	Computer Functions
Week 3	Introduction of Computer System Components
Week 4	Computer System Components: Processor Components
Week 5	Computer System Components: Internal Memory (RAM and ROM)

Week 6	Computer System Components: Storage Devices (secondary storage)
Week 7	Internal Hardware Computer Components: Power supply, Computer case, Internal cables
Week 8	Introduction of External Hardware Computer Components - Input Devices
Week 9	External Hardware Computer Components - Output Devices
Week 10	Introduction of System Software
Week 11	Computer Software - Operating Systems
Week 12	Computer Software - Utility Programs
Week 13	Application Software – M.S. Word
Week 14	Application Software – M.S. Excel
Week 15	Application Software – M.S. PowerPoint
Week 16	Preparatory week before the final exam.

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
	Material Covered
Week 1	Lab1_ Introduction of Computer Organization to understand the internal workings of the CPU and memory.
Week 2	Lab2_ Computer architecture simulations: Use simulation tools to explore different computer architectures, such as the von Neumann architecture or pipelined processors.
Week 3	Lab3_ Memory hierarchy experiments: Measure and analyze the performance of different levels of cache memory and understand their impact on system performance.
Week 4	Lab4_ System Software Lab: Operating system installation and configuration: Install and configure different operating systems (e.g., Windows) on virtual machines or physical hardware.
Week 5	Lab5_ Document processing and formatting using M.S. Word: Create and format documents using word processing software, applying different styles, headers, footers, and page layouts.
Week 6	Lab6_ Spreadsheet analysis and data manipulation using M.S. Excel: Perform data analysis tasks, such as sorting, filtering, and creating formulas and functions in spreadsheet software.
Week 7	Lab7_ Presentation design and delivery using M.S. PowerPoint: Create visually appealing presentations with multimedia elements and practice delivering effective presentations.
Learning and Teaching Resources	

مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	1. Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy. 2. "Computer Organization and Architecture: Designing for Performance" by William Stallings. 3. "Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne. 4. "Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne.	Yes
Recommended Texts	1. "Structured Computer Organization" by Andrew S. Tanenbaum and Todd Austin. 2. "Operating Systems: Internals and Design Principles" by William Stallings. 3. "Exploring Microsoft Office 2019" by Mary Anne Poatsy, Keith Mulbery, Cynthia Krebs, and Lynn Hogan.	No
Websites	https://ccms.tu.edu.iq/csd/electronic-lectures/409-stage1-8.html	

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	Discrete Structure			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TU010101108				
ECTS Credits	6				
SWL (hr/sem)	150				
Module Level		1	**Semester of Delivery**		1
Administering Department		Computer	**College**	CCSM	
Module Leader	Salwa Khalid Abdulateef		**e-mail**	Khalid.salwa@tu.edu.iq	
Module Leader's Acad. Title		Assistant Professor	**Module Leader's Qualification**		MSc.
Module Tutor	None		**e-mail**		
Peer Reviewer Name		None	**e-mail**		
Scientific Committee Approval Date		01/9/2025	**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1- Discrete structures provides the mathematical foundations for many courses including data structures and algorithm, Compilers, Automata theory and formal languages, operating systems theory and many other subjects. 2- Discrete structures are the abstract mathematical structures used to represent discrete object and relations lies between those objects. 3- Students' realization of the basic concepts of discrete structure, such as mathematics logic, graphs. 4- Knowing the models of discrete structures and how to create them. 5- Developing students' ability to deal with transformations and their applications in constructing structures. 6- Giving the student the necessary experience to deal with the relations and applications Giving students the necessary experience to solve some of the functions as linear and factorials.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	**1** Subject-specific skills: - Learn to prove the correctness and accuracy of the given issue, whether it is solvable or not, before starting to think about solving it. **2** Thinking skills :Giving the learner the skill to use logical hypotheses in building accurate software. **3** Giving the learner the skills to have the ability to build relationships between components, models and theoretical structures with algorithms and computer programs. **4** Enabling students to continue self-development after graduation. **5** Making the learner well acquainted with all types of logical deductive proof and types of proof by other methods. **6** Build basic causal skills in creating and validating algorithms and programs. - **7** Building skills to analyze and solve some important issues and the approximate time to solve them. - **8** Build skills on how to choose the appropriate solutions for some issues and designate the best algorithms to solve them
Indicative Contents المحتويات الإرشادية	Indicative content includes the following: Part A – Mathematics Logic: Foundations of Logic:

	Overview Proposition, compound proposition, How to Built a Truth Table, Logical operators, Bit operators and Translation English Sentences into Propositional Logic and Vice Versa with some examples. [16 hours] Part B- Set theory Definition of sets, Sets of numbers such as standard numerical , power set, cardinality, cartesian products of set, Venn Diagram, set operations, Algebra of sets with some examples [16 hours] Part C- Relations Definition of Relation, Graphical Representation of Relation, Properties of relations such as reflexive, symmetric and transitive with examples. [12 hours] Part D- Functions Definition of Function and examples, Types of function as one to one, onto, representation of function , application of functions with exampes. [16 hours)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	• The teacher gives detailed theoretical lectures • The teacher requests periodic reports on the basic topics of the subject • The student is also assigned to self-read and to give the student a certain period to inquire and discuss the topics he has read. • Solve practical examples Evaluation modalities 1-Daily exams with practical and scientific questions. 2- Participation scores for difficult competition questions among students. 3- Setting grades for homework and the reports assigned to them. 4- Quarterly exams for the academic curriculum in addition to the mid-year exam and the final exam.

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	87	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	5.8
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	15% (15)	4, 10,12	LO #1-5 , LO# 9 - 12
	Assignments	2	10% (10)	2, 12	LO # 3, 4, LO#8 -10
	Projects / Lab.				
	Report	1	15% (15)	12	LO # 8, 10 and 12
Summative assessment	**Midterm Exam**	2 hr	10% (10)	7	LO # 1-8
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	Mathematical logic
Week 2	Logical operators
Week 3	Logical operators
Week 4	Logical Equivalences, Compound Propositions Classification:
Week 5	Examples, Set of theory, Properties of set
Week 6	Sets of Number, Sets and elements, subsets
Week 7	**Mid Exam**
Week 8	Set's Algebra,
Week 9	Set's Algebra with examples
Week 10	relations
Week 11	Properties of relations
Week 12	Properties of relations with examples
Week 13	Review of Functions
Week 14	Types of Functions
Week 15	mathematics functions
Week 16	**Preparatory week before the final exam.**

Delivery Plan (Weekly Lab. Syllabus): There is no Lab activities المنهاج الاسبوعي للمختبر:لا توجد فعاليات مختبرية	
	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	Theory and problems of Discrete mathematics, by Seymour Lipschutz & Marc Lars Lipson, Schaum's Outline Series, third edition 2007 Discrete Mathematics and Its Applications, Seventh Edition, Kenneth H. Rosen, AT&T Laboratories, 2012	Yes
Recommended Texts	• Mathematical foundation of computer science, Y.N. Singh, 2005 • Discrete structures, Amin Witno, Revision Notes and Problems 2006, www.witno.com • Discrete mathematical structures for computer science by Bernard Kolman & Robert C. Busby	No
Websites	• http://en.wikibooks.org/wiki/Discrete_mathematics/Set_theory	

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.

Differential and Integral Calculus

Module Information معلومات المادة الدراسية					
Module Title	Differential and Integral Calculus			**Module Delivery**	
Module Type	Supportive			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TU010101110				
ECTS Credits	6				
SWL (hr/sem)	150				
Module Level		1	**Semester of Delivery**		1st
Administering Department		Computer sciences	**College**	CCSM	
Module Leader	Suad Mohammed shakor		**e-mail**		
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		Master

Module Tutor		**e-mail**	
Peer Reviewer Name	Mohammed Aktham	**e-mail**	
Scientific Committee Approval Date	01/09/2025	**Version Number**	1.0

Relation with other Modules
العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	Calculus II, Advanced Calculus	**Semester**	3

Module Aims, Learning Outcomes and Indicative Contents
أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Aims أهداف المادة الدراسية	The goal of studying differential calculus at university is to enable students to gain a deep understanding of this fundamental element of mathematics and its applications in different fields. By studying differential calculus, students learn how to calculate derivatives and understand the concept of a derivative as the instantaneous rate of change of a function. Students can apply the concepts of calculus to solve practical problems, analyze the behavior of functions, determine critical points, least and largest values of functions, and estimate changes of variable quantities. In addition, the study of differential calculus provides a foundation for the study of other topics in mathematics, science, and engineering, such as integration, calculus in multiple variables, and the solution of differential equations. Learning differential calculus aims to develop students' analytical thinking and mathematical reasoning capabilities and provide them with powerful mathematical tools to deal with complex technical and scientific problems.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Deep understanding of concepts: Students are expected to gain a deep understanding of basic concepts in differential calculus. Students should be able to understand the concept of a derivative and its meaning as the rate of change of a function, as well as the concept of inverse differential and integration of functions. 2. Numerical and Application Skills: Students should acquire strong skills in calculating and using derivatives in solving applied calculus problems. They should be able to compute the derivative of a variety of functions and apply it in analyzing the behavior of functions, identifying critical points, and estimating absolute values and variable ratios. 3. Analytical Thinking: By studying differential calculus, students are expected to develop abilities in analytical thinking and mathematical reasoning. They should be able to analyze mathematical problems and draw conclusions based on learned mathematical concepts and tools. 4. Applications in Other Fields: Students should have the ability to apply the

	concepts of differential calculus in other fields such as science, engineering, and economics. They learn how to represent real phenomena by functions and use differential calculus to analyze these phenomena and derive practical results. 5. Use of Technology: Students should have the ability to use appropriate technology such as mathematical calculation programs and
Indicative Contents المحتويات الإرشادية	Indicative content includes the following: 1. Introduction to calculus: includes the definition of a derivative and the concept of a derivative as the instantaneous rate of change of a function. Students are exposed to the basic rules of calculus and related concepts. 2. The basic rules of differentiation: It includes studying the basic rules of differentiation such as the rule of differentiation rules, the rule of differentiation of constants, the rule of differentiation of forces, and other rules of differentiation of known functions. 3. Higher Derivatives: Students learn how to calculate higher derivatives, how to work with recursive differentiation, and how to use differential rules related to it. 4. Applications in differential calculus: Students explore the practical applications of calculus in different fields such as physics, engineering, and computer science. Practical examples of solving various differential problems are presented. 5. Relative and Total Differential: Students learn the concept of relative differential and total differentiation and how to calculate them. They are exposed to its applications in analyzing the behavior of functions and estimating variable changes. 6. Practical Applications of Calculus: Students are introduced to the use of differential calculus in solving problems in mathematical modeling, economic analysis, statistics, and other fields.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Student interaction: Active participation and interaction between students and the lecturer or teacher is encouraged. Small group discussions or collaborative sessions can be organized to solve various differential problems. Technology, such as online forums or distance learning tools, can be used to encourage communication and collaboration among students. 2. Practical Application and Projects: The course should include practical activities and application projects that allow students to apply differential concepts and skills in real-world contexts. For example, teams can be formed to solve multidimensional differential problems or applications in fields such as engineering and medical science. 3. Use of Technology: Calculus software and mathematical applications can be used to enhance interaction and interactive learning. Students can use

	graphing software or computer mathematics programs to analyze functions and graph their curves. 4. Provide examples and practical exercises: A wide range of examples and practical exercises covering various differential calculus concepts should be provided. Students can practice solving the exercises

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	77	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	5.13
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	73	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	4.86
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)	5, 10	LO #1-3 , LO# 4 - 5
	Assignments	2	10% (10)	2, 12	LO #1-3 , LO# 4 - 5
	seminar	1	10% (10)	9	LO # 1-5
	Report	1	10% (10)	12	LO #2-5
Summative assessment	**Midterm Exam**	2 hr	10% (10)	7	LO # 1-4
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to function, domain, range Invers functions, even and odd function
Week 2	. Graph the functions
Week 3	Limits and continuous
Week 4	Exponential Functions , Logarithm Functions, Trigonometric functions.
Week 5	Derivatives, Rules of differentiation, Applications of Derivatives.

Week 6	The mean value theorem
Week 7	**Mid-term exam**
Week 8	The derivative and extrema
Week 9	Derivatives of Exponential Functions , Logarithm Functions
Week 10	Derivatives of Trigonometric functions, Derivatives of inverse functions
Week 11	Integration
Week 12	The mean value theorem for integrals
Week 13	basic application of integration
Week 14	Area , volume
Week 15	Arc length
Week 16	**Preparatory week before the final exam.**

Delivery Plan (Weekly Lab. Syllabus): There is no Lab activities المنهاج الاسبوعي للمختبر:لا توجد فعاليات مختبرية	
	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. Courant, R., John, F., Blank, A. A., & Solomon, A. (1965). *Introduction to calculus and analysis* (Vol. 1). New York: Interscience Publishers. 2. Tall, D. (1996). Functions and calculus. *International handbook of mathematics education*, *1*, 289-325. 3. Tall, D. (1996). Functions and calculus. *International handbook of mathematics education*, *1*, 289-. 4. Marsden, J., & Weinstein, A. (1985). *Calculus I*. Springer Science & Business Media.	No

	5. *Thomas' Calculus*, Early Transcendental, 12th ed. 6. Calculus and Analytic Geometric, Durfee. W.H ,1971 New York (3).	
Recommended Texts	Grossman, Stanley I. *Calculus*. Academic Press, 2014.	No
Websites	https://books.google.iq/books?hl=ar&lr=&id=0aziBQAAQBAJ&oi=fnd&pg=PP1&dq=calculus+book&ots=a1k4tINdCZ&sig=tmAQQ_yHi9mTDBLcx-qi7hy9uo8&redir_esc=y#v=onepage&q=calculus%20book&f=false	

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	Elective			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TU010101101				
ECTS Credits	2				
SWL (hr/sem)	50				
Module Level		1	**Semester of Delivery**		1
Administering Department		Computer	**College**	CCSM	
Module Leader	Ahmed fayeq saber		**e-mail**	Ahmed.f.saber@tu.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		M.Sc.
Module Tutor	None		**e-mail**	------	
Peer Reviewer Name		Ayham Mahmood	**e-mail**		
Scientific Committee Approval Date		1/9/2025	**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	0
Co-requisites module	None	**Semester**	0

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To enable the students to communicate effectively and appropriately in real life situation.

	2- To develop and integrate the use of the four language skills i.e., Reading, Listening, Speaking and Writing. 3. To use English effectively for study purpose across the curriculum. 4. The ability to understand meaning of words, phrases and sentences in context. 5. The ability to speak and pronounce English Correctly and intelligibly 6. The ability to write English correctly and master the Mechanics of writing; the use of correct punctuation marks and capital letters. 7- To acquire the ability to use a suitable dictionary to understand labels, simple notices and written instructions. 8- To enable the students to know the contemporary strategies in teaching and learning English language.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Demonstrate through face-to-face conversations comprehension of simple words and phrases used in common everyday context. 2.. Knowing the basic rules of the English language. 3- Familiarity with the four English language skills: listening, reading, speaking and writing. 4. Determine the negative effects of the mother tongue on the English language. 5. Interpreting the texts in different contexts. 6- Generate simple sentences containing learned vocabulary and using appropriate grammatical structures 7- Express awareness of social and environmental issues. 8- Acquire varied range of vocabulary; understand increased complexity of sentence structures both in reading and writing. 9- Obtaining a core competency such as, developing the ability to express student's thoughts orally and in- writing in a meaningful way in English language. 10- Enable students to communicate in English more confidently and effectively in their work or study environment.
Indicative Contents المحتويات الإرشادية	**Indicative content** - Teaching students how to communicate each other in English by using the four skills: speaking, listening, reading and writing. The use of different examples from the everyday life, dialogues, conversations and the weekly compositions are useful for more development. [50 hours]

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1- This course is characterized by the fact that it needs a special strategy that depends mainly on the development of English language and its skills. It also depends on previous courses in real analysis, situation, and some imagination. Teaching is mainly based on the home works that are given at the end of each week, and the student notes the interdependence between the serial topics of this course. In addition, to assigning the student (or a group

	of students) to make one seminar for the purpose of training in the use of scientific resources and the method of writing a subject in English language. 2- The purpose of this module is to develop students' linguistic ability by focusing on the key skills of reading, writing, speaking and listening, to encourage students to become independent learners and to introduce them to strategies and skills to enable them to cope with the demands, both academic and cultural, of undergraduate study in an English-speaking environment.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	32	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	18	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	1.2
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, 10	LO #1-3 , LO# 4-7
	Assignments	2	10% (10)	2, 12	LO # 2, 4, LO# 5-8
	Seminar	1	10% (10)	11	LO # 1-8
	Report	1	10% (10)	12	LO # 1-8
Summative assessment	**Midterm Exam**	2 hr	10% (10)	7	LO # 1-8
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	**An introduction to English language.**
Week 2	**Unit (1) Hello am/ are/ is, my/your · This is … • How are you? • Good morning! What's this in English? • Numbers • Plurals**

Week 3	**Unit (2) Your World** Countries • he/she/ they, his/her • Where's he from? fantastic/ awful/ beautiful · Numbers 11-30
Week 4	**Unit (3) All about You** Jobs • am/are/ is • Negatives and questions • Personal information • Social expressions
Week 5	**Unit (4) Family and Friends** our/their • Possessive 's • The family • has/have • The alphabet.
Week 6	**Unit (5) The Way I live** Sports/ Food/ Drinks • Present Simple - I/you/ we/ they • a/an Languages and nationalities • Numbers and prices.
Week 7	**Unit (6) Every day** the time • Present Simple-he/she · always/sometimes/never Words that go together • Days of week.
Week 8	**Unit (7) My favorites** Question words • me/him/us/them • this/that Adjectives • Can I.?
Week 9	**Unit (8) Where I live** Rooms and furniture • There is/ are • Prepositions • Directions
Week 10	**Unit (9) Times past** Saying years • was/were born • Past Simple - irregular verbs • have/do/go • When's your birthday?
Week 11	**Unit (10) We had a great time!** Past Simple - regular and irregular • Questions and negatives • Sport and leisure • Going sightseeing.
Week 12	**Unit (11) I can do that!** can/can't · Adverbs • Adjective + noun • Everyday problems.
Week 13	**Unit (12) Please and thank you** I'd like - some/any • In a restaurant • Signs all around.
Week 14	**Unit (13) Here and now** Colors and clothes • Present Continuous • Opposite verbs • What's the matter?
Week 15	**Unit (14) It's time to go!** Future plans • Grammar revision Vocabulary revision • Social expressions.
Week 16	**Preparatory week before the final exam.**

Delivery Plan (Weekly Lab. Syllabus): There is no Lab activities المنهاج الاسبوعي للمختبر:لا توجد فعاليات مختبرية	
	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- Oxford Headway plus for Beginners. 2- New Headway English Course (2002) by Julia Starr Keddle.	Yes
Recommended Texts	English for Everyone	No
Websites	https://www.unionlearn.org.uk/english-and-maths-learning-resources-and-tools	

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	Democracy and Human rights	Module Delivery

Module Type	E	☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☒ Seminar
Module Code	TU010101102	
ECTS Credits	2	
SWL (hr/sem)	50	

Module Level	UGI	Semester of Delivery	1
Administering Department	Computer Science	College	CCSM
Module Leader	Saad Hussein ali	e-mail	Saad.h.ali@tu.edu.iq
Module Leader's Acad. Title	Assistant Legturer	Module Leader's Qualification	Master
Module Tutor		e-mail	
Peer Reviewer Name	Mohammed Aktham Ahmed	e-mail	
Scientific Committee Approval Date	22/10/2025	Version Number	1.0

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	• The ability to understand the basic concept of democracy. • The ability to understand the historical origins of the two concepts. • Recognizing the sources characteristics and features of democracy. • Gaining insight into democracy in Islam. • Understanding the impact of technological development.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	
Indicative Contents	

المحتويات الإرشادية	

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	**The teaching and learning strategy was designed to ensure that students acquire comprehensive knowledge coveing the course curriculum thereby achieving the main objective of the progrom- enabling students to grasp and understand the fundamental concepts of human rights and democracy as well as to become familiar with the sources safeguards and international conventions related to these concepts this knowledge aims to be utilized in addressing negative phenomena within society and in maintaining social stability and peace.**

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 المن	2.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	17	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	1.1
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)	4 , 10	2-5, 7-9
	Assignments	4	20% (20)	3,7,10,13	1, 3, 5,8,9,10
	Projects	1	20% (20)	14	18
	Report	1		9	
Summative assessment	Midterm Exam	2 hr	10% (10)		
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	
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	

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	
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		No
Recommended Texts		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

.

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					
ECTS Credits	6				
SWL (hr/sem)	148				
Module Level	3		**Semester of Delivery**	first	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	**Mohanad Dawood Salman**		**e-mail**	mohanaddawoodalroomi@tu.edu.iq	
Module Leader's Acad. Title	**Lecturer**		**Module Leader's Qualification**	Master	
Module Tutor	**Luay Ibrahim Khalaf**		**e-mail**	luay.i.khalaf@tu.edu.iq	
Peer Reviewer Name	Mahammed Aktham		**e-mail**		
Scientific Committee Approval Date	1/9/2025		**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	To provide students with a solid foundation in the principles, methodologies, and tools of software engineering. To enable students to design, develop, and manage large-scale software systems systematically.

	To introduce modern software development models, from requirements analysis to deployment and maintenance. To equip students with teamwork, project management, and documentation skills for professional software development. To raise awareness of software quality assurance, ethical issues, and industry standards.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Explain fundamental concepts, processes, and life cycle models in software engineering. Analyze user requirements and specify functional and non-functional requirements. Apply software design principles (e.g., modularity, abstraction, design patterns). Develop software systems using structured and object-oriented methodologies. Employ project management techniques, including planning, scheduling, and risk management. Evaluate and apply software testing, verification, and validation methods. Use collaborative tools and version control systems in software projects. Demonstrate awareness of professional ethics, software quality standards, and emerging practices in the field.
Indicative Contents المحتويات الإرشادية	• **Introduction to Software Engineering**: Nature, scope, challenges. • **Software Development Life Cycle (SDLC)**: Waterfall, Agile, Spiral, and hybrid models. • **Requirements Engineering**: Elicitation, analysis, documentation, validation. • **Software Design**: Architectural styles, UML diagrams, design principles, design patterns. • **Implementation Issues**: Coding standards, documentation, integration. • **Software Testing & Quality Assurance**: Unit testing, integration testing, system testing, test automation. • **Project Management**: Estimation (cost, time), scheduling (Gantt, PERT), risk management, team roles. • **Configuration Management**: Version control (Git), continuous integration. • **Software Maintenance & Evolution**: Legacy systems, refactoring, re-engineering. • **Professional Issues**: Ethics, intellectual property, standards (IEEE, ISO). • **Emerging Trends**: DevOps, cloud-based development, model-driven engineering.

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم	
Strategies	**Lectures & Tutorials**: Deliver theoretical foundations and problem-solving methods. **Laboratory Sessions**: Hands-on practice with CASE tools, UML modeling, and project management software. **Team-Based Projects**: Develop a complete software system through group collaboration. **Case Studies**: Analysis of real-world software failures and successes. **Workshops & Seminars**: Discussions on Agile methods, DevOps, and modern practices. **Peer Assessment & Presentations**: Encourage teamwork, communication, and critical evaluation. **Industry Linkage**: Guest talks from software practitioners and simulated real-world projects.

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

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)		
	Assignments	4	20% (20)		
	Projects	1	20% (20)		
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)		
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Introduction to Software Engineering – Principles, software process, life cycle.
Week 2	Software Development Models – Waterfall, Agile, Spiral, V-Model.
Week 3	Software Project Management – Planning, estimation, scheduling.

Week 4	Software Requirements Engineering – Elicitation, analysis, specification, validation.
Week 5	System Modeling – Use cases, UML diagrams, behavioral & structural modeling.
Week 6	Software Design Principles – Modularity, abstraction, design patterns.
Week 7	Architectural Design – Client-server, layered, MVC, microservices.
Week 8	**Midterm Exam** (Covers first half of course).
Week 9	Implementation Issues – Coding standards, code review, version control.
Week 10	Software Testing – Levels of testing, unit, integration, system, acceptance.
Week 11	Advanced Testing – Black-box, white-box, automated testing tools.
Week 12	Software Maintenance & Evolution – Types, costs, strategies.
Week 13	Software Quality Assurance – Metrics, reviews, standards (ISO, CMMI).
Week 14	Emerging Trends – DevOps, continuous integration, cloud-based SE.
Week 15	Revision & Final Exam Preparation.

Delivery Plan (Weekly Lab. Syllabus):

المنهاج الاسبوعي للمختبر:

Week No.	Material Covered
Week 1	Introduction to Software Engineering tools and Git/GitHub basics.
Week 2	Writing Software Requirement Specification (SRS) for a case study.
Week 3	Drawing UML use case diagrams.
Week 4	Drawing UML class and sequence diagrams.
Week 5	Modeling system architecture with deployment & component diagrams.
Week 6	Designing database schema (ER diagrams) for the case study project.
Week 7	Applying design patterns (Factory, Singleton, Observer).
Week 8	**Midterm Practical Test.**
Week 9	Implementing a small module of the project (using Java/Python/C#).
Week 10	Writing unit tests (JUnit/PyTest/NUnit).
Week 11	Performing integration testing on the project.
Week 12	Code quality check using static analysis tools.
Week 13	Software maintenance exercise – Refactoring legacy code.
Week 14	Applying Agile/Scrum with project management tools (Jira/Trello).
Week 15	Final project presentation & practical revision.

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	**Available in the Library?**
Required Texts	Software Engineering /Tenth Edition /Ian Sommerville	yes
Recommended Texts		
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 Architecture			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	TU010102002				
ECTS Credits					
SWL (hr/sem)	60				
Module Level	2		**Semester of Delivery**	3rd	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Firas Layth Khaleel		**e-mail**	Firas_Layth@tu.edu.iq	
Module Leader's Acad. Title	Assoc. Prof. Dr.		**Module Leader's Qualification**	PhD	
Module Tutor	Firas Layth Khaleel		**e-mail**	Firas_Layth@tu.edu.iq	
Peer Reviewer Name	Dr. Mahammed Aktham		**e-mail**		
Scientific Committee Approval Date	01/09/2025		**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	**The course aims to elucidate the fundamentals of computer organization and the fundamental and advanced architectural concepts. The text commences with the fundamental architecture and design of a basic digital computer and presents a straightforward register transfer language to delineate various computational activities, culminating in parallel systems and message forwarding via routing.** **This course is to establish a robust foundation for students to comprehend contemporary computer architecture. The course is organized around many**

	primary topics in computer architecture. The themes encompass several components of a computer, including memory, CPU, and input/output devices.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	**Upon successful completion of this module, students should be able to:** **This module provides a comprehensive foundation in computer architecture, beginning with its historical evolution and progressing to its modern implementations. Students will achieve a deep understanding of the fundamental components, elements, and structures of computer systems, including the central processing unit (CPU), memory hierarchies, bus systems, and input/output mechanisms. This knowledge is framed within a practical context, enabling learners to explain how these hardware solutions are designed and selected to meet the specific needs of individuals or organizations. The curriculum ensures that upon completion, students can describe the intricate operations of the CPU, the design principles behind memory systems for optimal performance, and the concepts of parallel processing, thereby connecting theoretical architecture to tangible technological solutions.** **Building upon this foundational knowledge, the module develops analytical and design-oriented skills. Students will learn to dissect various low-level programming language instructions, understanding their syntax and semantics to solve specific programming problems at a hardware-near level. This analytical capability is directly applied to the design and integration of computer systems. Learners will be equipped to combine computer components and interconnections, making informed decisions to propose appropriate mathematical and architectural solutions. The goal is to balance critical engineering trade-offs, achieving optimal cost-effectiveness and performance targets for a given set of requirements, thus moving from simple recognition of parts to active system synthesis.** **The module then focuses on the practical application of techniques to enhance system functionality and performance. Students will employ various methods and tools to understand how data is transferred between different units of computer architecture, such as between the CPU and memory or I/O devices, and will explore strategies to improve the efficiency of these transfers. A core technical skill developed is the ability to perform arithmetic and logic operations to design interface instructions. This involves creating the low-level code that acts as a bridge between computer programs and the hardware, enabling software to interact directly and efficiently with physical components, which is crucial for systems programming and driver development.** **Finally, the curriculum emphasizes professional competencies and holistic system operation. Students will learn to select and integrate both appropriate software features and hardware components, operating them effectively within an individual or organizational context. Furthermore, the module places a strong emphasis on collaborative work, teaching students how to communicate effectively within a team environment. A key practical outcome is the development of programmable skills in assembly language, where students will not only write code but also explain and demonstrate how such low-level programming facilitates direct interaction with the**

	computer's hardware, thereby unifying the theoretical, practical, and interpersonal aspects of computer architecture.
Indicative Contents المحتويات الإرشادية	**• A Brief History of Computers - Computer Generations** **• General architecture of computer** **• Overview of 8085 Microprocessor** **• 8086 Microprocessor** **• 8086 instruction set and Assembly language program** **• Multiprocessors:** **• Parallel Processors** **• Four Decades Of Computing** **• Flynn Calcification Of Computer Architecture** **• Interconnection Networks** **• Interconnection Networks Taxonomy** **• Performance Analysis Of Multiprocessor Architectures** **• Scalability Of Parallel Architectures** **• Benchmark Performance** **• Shared Memory Architecture** **• Classification Of Shared Memory Systems** **• Bus-Based Symmetric Multiprocessors** **• Basic Cache Coherency Methods** **• SNOOPING PROTOCOLS** **• Message Passing** **• Routing In Message Passing Networks** **• Routing for Broadcasting and Multicasting** **• Routing Potential Problems** **• Switching Mechanisms In Message Passing**

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	**- Interactive Lectures** **- Dialogue and discussion** **- Brainstorming** **- Solving problems** **- Projects, tasks and costs projects**

	- Self-learning - Learning Cooperative - Exchanging experiences among colleagues.

Student Workload (SWL)
الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

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

Module Evaluation
تقييم المادة الدراسية

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

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	A Brief History of Computers - Computer Generations, General architecture of computer, Overview of 8085 Microprocessor
Week 2	8086 Microprocessor, 8086 instruction set and Assembly language program
Week 3	Multiprocessors, Parallel Processors, Four Decades Of Computing, Flynn Calcification Of Computer Architecture
Week 4	Interconnection Networks, Interconnection Networks Taxonomy
Week 5	Exam
Week 6	Performance Analysis Of Multiprocessor Architectures

Week 7	Scalability Of Parallel Architectures, Benchmark Performance
Week 8	Shared Memory Architecture, Classification Of Shared Memory Systems
Week 9	Bus-Based Symmetric Multiprocessors, Basic Cache Coherency Methods, SNOOPING PROTOCOLS
Week 10	Exam
Week 11	• Message Passing • Routing In Message Passing Networks
Week 12	• Routing for Broadcasting and Multicasting
Week 13	• Routing Potential Problems
Week 14	• Switching Mechanisms In Message Passing
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	From Assembly to Abstraction - Basic C++ Practical Topic: Setting up the C++ IDE and understanding the transition from low-level to high-level programming.
Week 2	The Foundation of OOP - Classes and Objects Practical Topic: Modeling Real-World Entities.
Week 3	The Foundation of OOP - Classes and Objects Practical Topic: Modeling Real-World Entities.
Week 4	Controlling Access and State - Encapsulation and Constructors Practical Topic: Implementing Data Hiding and Initialization.
Week 5	Practical Exam 1
Week 6	Polymorphism in Action - Function Overloading and Operator Overloading Practical Topic: One Interface, Multiple Implementations.
Week 7	Polymorphism in Action - Function Overloading and Operator Overloading Practical Topic: One Interface, Multiple Implementations
Week 8	Inheritance - Creating Hierarchies Practical Topic: Modeling "Is-A" Relationships and Code Reusability.
Week 9	Inheritance - Creating Hierarchies Practical Topic: Modeling "Is-A" Relationships and Code Reusability.
Week 10	Practical Exam 2
Week 11	Constructor 1
Week 12	Constructor 2
Week 13	Destructor 1
Week 14	Destructor 2
Week 15	Final Project Demonstration and Review

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts		Yes
Recommended Texts	***Advanced Computer Architecture And Parallel Processing* By H El-Rewini, M Abd-El-Barr**	No
Websites	https://books.google.iq/books?hl=en&lr=&id=7JB-u6D5Q7kC&oi=fnd&pg=PR7&dq=ADVANCED+COMPUTER+ARCHITECTURE+AND+PARALLEL+PROCESSING&ots=4uomx3qv08&sig=9uU2ZtNWG9L5sMLmkziHlYp_HaQ&redir_esc=y#v=onepage&q=ADVANCED%20COMPUTER%20ARCHITECTURE%20AND%20PARALLEL%20PROCESSING&f=false	

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	Data Structure			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TU010102005				
ECTS Credits					
SWL (hr/sem)	148				
Module Level	2		**Semester of Delivery**	1nd	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	**Mohammed Basim Omar**		**e-mail**	**mohammed.b@tu.edu.iq**	
Module Leader's Acad. Title	Asst. lect.		**Module Leader's Qualification**		
Module Tutor			**e-mail**		
Peer Reviewer Name	Mohamed Aktham		**e-mail**		
Scientific Committee Approval Date	26/08/2024		**Version Number**		

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims	1. Learning different data structures

أهداف المادة الدراسية	2. Learning how to choose the best data structure for your algorithm. 3. learn how to deal with your problem, building its algorithm and fitting the best data structures to it. 4. Building the student scientifically and qualifying him to work in the field of computer science. 5. Building and preparing the student psychologically to play his role as a reliable programmer in this field. 6. A general definition of the subject of data structures and how to store data in memory. 7. The difference between linear and non-linear data structures. 8. A detailed explanation of algorithms and how to program them. 9. Teaching the student to calculate the complexity time of the algorithm. 10. Encouraging the student to be creative and think about the specialization projects and keep pace with the development taking place in this field.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Knowledge of the basic concepts of data structures. 2. The student learns about the importance of general concepts of algorithms. 3. Identify the most important algorithms used to organize data in memory. 4. Study the most important factors that affect the speed of algorithm execution
Indicative Contents المحتويات الإرشادية	1. Review of Basic Programming Concepts: - Recap of fundamental programming concepts, including variables, data types, control structures, and functions. 2. Arrays: - Multidimensional arrays - Array manipulation techniques - Searching and sorting algorithms 3. Strings: - String manipulation and operations - String handling functions 4. Pointers: - Introduction to pointers and their usage - Memory addresses and pointer arithmetic - Pointers to arrays 5. Tree: - tree handling concepts - Searching from and writing to tree 6. Graph - Graph handling concepts

	- explain types of graphs - Graph Representation

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain data structure concepts, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning. 4. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 5. Office Hours and Individual Support: The instructor should be available for

	individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

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

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	5-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Introduction to Data Structure.
Week 2	Algorithms and Complexity.
Week 3	Array in Data Structure & Represent arrays in memory.
Week 4	Pointer.
Week 5	Stack in Data structure.
Week 6	Expression Parsing in Data structure.
Week 7	**First exam.**
Week 8	Queue in Data structure.
Week 9	Circular queue in Data structure.

Week 10	LinkedList in Data structure.
Week 11	Introduction Tree.
Week 12	binary tree & binary search tree.
Week 13	Graph.
Week 14	Sorting.
Week 15	Hashing.

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Re-explaining the basics of programming to the student in C++.
Week 2	Introduction on Array programming
Week 3	Explain how stack types work in programming [1].
Week 4	**Explain how stack types work in programming [2].**
Week 5	Explain how queue types work in programming [1].
Week 6	Explain how queue types work in programming [2].
Week 7	Start Of Linked List programming.
Week 8	Add Node & Delete Node from Linked list [1]
Week 9	Add Node & Delete Node from Linked list [2]
Week 10	How Work Binary Search Programming.
Week 11	Binary Search with Iterative Programming.
Week 12	Binary Search with Recursive Programming.
Week 13	Selection Sort programming.

Week 14	Bubble Sort Programming.
Week 15	Comprehensive review in Programming.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	• Vinu V Das, *Principles of Data Structures Using C and C++* • Introduction to Algorithm, third Edition, Thomas H. Cormen Algorithms, fourth edition, Robert Sedgewick and Kevin Wayne.	Yes
Recommended Texts	• Ramesh Vasappanavara, Anand Vasappanavara, *Data Structures using C by practice*, • D.S. MALIK, *Data Structures Using C++*, 2nd Edition, • Robert L. Kruse, Alexander J. Ryba, *Data Structures and Program Design in C++*,	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	Database Basics	**Module Delivery**	
Module Type	Core	☐ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	TU010102016		
ECTS Credits	8		
SWL (hr/sem)	200		
Module Level	1	**Semester of Delivery**	1st
Administering Department	Computer Science	**College**	CCSM
Module Leader	Mustafa Lateef Fadhil	**e-mail**	Mustafa.l.fadhil@tu.edu.iq
Module Leader's Acad. Title	Assistant Lecturer	**Module Leader's Qualification**	master
Module Tutor		**e-mail**	
Peer Reviewer Name	Mahammed Aktham	**e-mail**	
Scientific Committee Approval Date	/ /2025	**Version Number**	1.0

Relation with other Modules

العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	Relational databases and structured query language	**Semester**	2

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To introduce students to the fundamental principles and concepts of database. 2. To familiarize students with the syntax and structure of the SQL server programming language. 3. To develop students' problem-solving skills for database and tables design. 4. To enable students to design, implement, and test programs using SQL server. 5. To provide students with hands-on experience in programming through practical exercises, assignments, and projects. 6. To promote the use of SQL query techniques for creating reusable and maintainable code. 7. To enhance students' ability to debug and troubleshoot programs effectively. 8. To develop students' communication skills in expressing programming concepts and solutions clearly and effectively. 9. To prepare students for SQL query language and real-world software development scenarios.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Demonstrate a solid understanding of the fundamental principles and concepts of databases. 2. Utilize the syntax and structure of the SQL query language to write well-structured and efficient code. 3. Apply problem-solving skills and algorithmic thinking to develop solutions for a variety of programming problems.

	4. Design, implement, and test programs using SQL server to solve specific tasks and challenges. 5. Utilize modular SQL query techniques to create reusable and maintainable code. 6. Debug and troubleshoot programs effectively using appropriate debugging techniques and tools. 7. Collaborate and work effectively in teams to complete database projects. 8. Communicate programming concepts, solutions, and ideas clearly and effectively, both orally and in written form. 9. Demonstrate a readiness to progress to more advanced programming courses or pursue a career in software development.
Indicative Contents المحتويات الإرشادية	1. Introduction to Computer Science: - Overview of computer science as a discipline - Key concepts and principles in computer science - Role of programming in computer science 2. Introduction to Databases and Database users: - Characteristics of the Database Approach - Actors on the Scene - Workers behind the Scene - Advantages of Using the DBMS Approach - A Brief History of Database Applications - When Not to Use a DBMS 3. Database System Concepts and Architecture: - Data Models, Schemas, and Instances - Three-Schema Architecture and Data Independence - Database Languages and Interfaces - The Database System Environment - Centralized and Client/Server Architectures for DBMSs - Classification of Database Management Systems 4. Basic SQL: - SQL Data Definition and Data Types - Specifying Constraints in SQL - Basic Retrieval Queries in SQL - INSERT, DELETE, and UPDATE Statements in SQL - Additional Features of SQL - More SQL: Complex Queries, Triggers, Views, and Schema Modification 5. Variables and Data Types: - Declaring and initializing variables - Fundamental data types (integers, varchar(255), date, …) - Working with constants and literals

	7. Control Structures: - Decision-making with if-else statements - Versatile statement (CASE) - Repetition with loops (while, do-while, for) - Handling user input and validation 8. Functions: - Function declaration and definition - Parameters and argument passing - Return values and function overloading - Scope and lifetime of variables

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain database concepts, SQL syntax, and techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing database best practices, and exploring real-world applications of database concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on database exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on database exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-

	solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of database programming concepts and encourage independent problem-solving. These assignments may involve implementing sql query, designing database systems and tables. 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding database programming concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	93	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	107	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	7.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation

تقييم المادة الدراسية

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

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction to Computer Science, database , data, tables
Week 2	Introduction to Databases and Database users:
Week 3	Database System Concepts and Architecture
Week 4	Centralized and Client/Server Architectures for DBMSs
Week 5	Classification of Database Management Systems
Week 6	Basic SQL syntax
Week 7	Additional Features of SQL syntax
Week 8	More SQL: Complex Queries, Triggers, Views, and Schema Modification
Week 9	Variables and Data Types
Week 10	Fundamental data types (integers, varchar(255), date, …)
Week 11	Flow Control (if – else)
Week 12	Loops (counter)
Week 13	Repetition with loops (while, do-while, for)
Week 14	building a computer science and mathematics database
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Using Microsoft SQL Server Management Studio, Creating database and table, writing sql query)
Week 2	Difference among (Microsoft access, Microsoft Excel and SQL server)
Week 3	Design (Database and Table) Flowcharts
Week 4	**Installing Microsoft SQL Server Management Studio**
Week 5	Running Examples on Creating Database and Table

Week 6	Running Examples on Basic Retrieval Queries in SQL
Week 7	Running Examples on INSERT, DELETE, and UPDATE Statements in SQL
Week 8	Running Examples on More SQL: Complex Queries, Triggers, Views, and Schema Modification
Week 9	Running Examples on Flow Control (if – else)
Week 10	Running Examples on Loops (counter)
Week 11	Running Examples on Repetition with loops (while, do-while, for)
Week 12	Running Examples on Functions
Week 13	Building a computer science and mathematics database
Week 14	Fixing problems in students' projects
Week 15	Applying instructor's feedback on students' projects

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Elmasri, R. (2021). Fundamentals of database systems seventh edition.	Yes
Recommended Texts		
Websites		

Grading Scheme
مخطط الدرجات

Group	Grade	التقدير	Marks (%)	Definition
Success Group	**A** - Excellent	امتياز	90 - 100	Outstanding Performance

(50 - 100)	**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	TU010102007		
ECTS Credits	6		
SWL (hr/sem)	148		
Module Level	2	**Semester of Delivery**	3rd
Administering Department	Computer Science	**College**	CCSM
Module Leader	Israa Rafaa Abdulkader	**e-mail**	Israa.R.Abdalkader@tu.edu.iq
Module Leader's Acad. Title	lecture	**Module Leader's Qualification**	Master
Module Tutor		**e-mail**	
Peer Reviewer Name	Mohamed Aktham	**e-mail**	
Scientific Committee Approval Date	30/08/2025	**Version Number**	1.0

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. Abstraction: hiding the implementation details and showing only functionality to the user. 2. Encapsulation: data hiding. 3. Polymorphism: the ability to take more than one form. 4. Inheritance: objects of one class acquire the properties of objects of another class.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Codes basic programs in java programming language 2. Introduction (Object Oriented Programming Characteristics, OOP Definition, OOP Concepts, Differences from Procedure Oriented Programming (POP)). 3. Classes and Objects (methods, properties). 4. Constructors and Destructors. 5. Inheritance (Single Inheritance, Multilevel Inheritance) and Polymorphism (Function Overloading, Operator Overloading, Virtual Function). 6. Abstract Classes and Methods, Sealed Classes and Interfaces, Generic types and methods and collection class.
Indicative Contents المحتويات الإرشادية	Basics of java Characteristics of Object-Oriented Programming Classes and Objects Working with Constructors Data Members Using Static Variables & Understanding Scope Overloading Inheritance Exceptions and Errors

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain programming concepts, java syntax, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of programming concepts and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using java . 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for

	improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	9-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Recap of fundamental programming concepts, including variables, data types, control structures Using Java language
Week 2	Introduction to method(function)
Week 3	Mathematical Functions and Class Character
Week 4	Introduction to String and methods

Week 5	Introduction (Object Oriented Programming Characteristics, OOP Definition, OOP Concepts, Differences from Procedure Oriented Programming (POP)).
Week 6	Classes and Objects (methods, properties).
Week 7	Polymorphism (Overloading and Overriding Methods) & Constructors.
Week 8	Inheritance (Single Inheritance, Multilevel Inheritance).
Week 9	Abstract Classes and Methods, Interfaces, Generic types and methods..
Week 10	**Midterm Exam**
Week 11	Introduction to Arrays (Linear arrays) and Searching and Sorting Linear Arrays
Week 12	Multidimensional Arrays and Square Arrays
Week 13	Array List Class
Week 14	Project Due (Students Presentations part1)
Week 15	Project Due (Students Presentations part2)

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Getting used to java netbeans Interfaces and practicing some commands
Week 2	Running Examples on method
Week 3	Running examples on Mathematical Functions and Class Character
Week 4	Running Examples on Strings
Week 5	Running Examples on Object Oriented Programming Characteristics
Week 6	Running Examples Classes and Objects (methods, properties).
Week 7	Running Examples on Overloading , Overriding Methods and Constructors.
Week 8	Running More Examples on Inheritance (Single Inheritance, Multilevel Inheritance).
Week 9	Running Examples on classes

Week 10	Running More Examples on Abstract Classes and Methods, Interfaces, Generic types and methods.
Week 11	Practicing Arrays further (Searching and Sorting)
Week 12	Running Examples on 2D and Square Arrays
Week 13	Running examples on Array List Class
Week 14	Running programs of advanced of class
Week 15	Answering students' questions and reviewing previous topics.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Herbert Scildt, Java The complete reference Eleventh edition	No
Recommended Texts		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 DESCRIPTOR
وصف مادة حرية و ديمقراطية

Module Information معلومات المادة الدراسية		
Module Title	Ba'ath Party Crimes in Iraq	**Module Delivery**
Module Type	غير أساسية (داعمة)	☒ Theory ☐ Lecture ☐ Lab ☐ Tutorial ☐ Practical
Module Code	TU010102008	
ECTS Credits	2	
SWL (hr/sem)	50	

		☒ Seminar	
Module Level	1	**Semester (s) offered**	2
Min number of students	15	**Max number of students**	100
Administering Department	Math	**College**	CCSM
Module Leader	Saad Hussein Ali	**e-mail**	Saad.h.ali@tu.edu.iq
Module Leader's Acad. Title	Assistant Lecturer	**Module Leader's Qualification**	Master's
Module Tutor	None	**e-mail**	None
Peer Reviewer Name	Mohamed. A. ahmad	**e-mail**	
Review Committee Approval	25/09/2024	**Version Number**	1.0

Relation With Other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	لايوجد	**Semester**	1
Co-requisites module	لايوجد	**Semester**	-
Module Aims, Learning Outcomes, Indicative Contents and Brief Description أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية مع وصف مختصر			
Module Aims أهداف المادة الدراسية	**1-To identify the crimes committed by the baath party** **2- To study how the nobie Islamic religion addresses human rights in both religious and worldly aspects** **3- The course examines the methods used by the baath regime to starve the people**		
Module Learning Outcomes مخرجات التعلم للمادة الدراسية			
Indicative Contents المحتويات الإرشادية			
Course Description			
Learning and Teaching Strategies استراتيجيات التعلم والتعليم			
Strategies	The teaching and learning strategy was designed to ensure that the student acquires comprehensive information covering the curriculum prepared for the subject in order to achieve the main objective course which focuses on enabling the student to grasp		

Student Workload (SWL) الحمل الدراسي للطالب			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل **In class lectures 30** **In class tests 3**	33	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	2.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	17	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	1.1
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	50		

Module Evaluation
تقييم المادة الدراسية

		Time (hr)	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	4	20% (20)	3, 5, ,7, 9,11,13,	LO #1, 2,3,....., 11
	Assignments (Homeworks)	6	15% (15)	2, 4, 6, 10,12,14	LO # 1, 2, 3, ...,11
	Discussions	7	5% (5)	Continuous	
Summative assessment	**Midterm Exam**	2	10% (10)	8	LO # 1-7
	Final Exam	3	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	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	
Week 16	

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	**وزارة التعليم العالي والبحث العلمي**	Yes
Recommended Texts	.	No
Websites	N/A	

APPENDIX:

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	مقبول بقرار	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required
Note:				

NB 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	Mobile Application Development	Module Delivery	
Module Type	Core	☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☒ Seminar	
Module Code	TU010103016		
ECTS Credits	6		
SWL (hr/sem)	148		
Module Level	3	Semester of Delivery	first

Administering Department	Computer Science	College	CCSM
Module Leader	Qabas Amer Hameed	e-mail	
Module Leader's Acad. Title		Module Leader's Qualification	
Module Tutor	Suad Mohammed Shkoor	e-mail	
Peer Reviewer Name		e-mail	
Scientific Committee Approval Date	1/9/2025	Version Number	1.0

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	• To provide students with theoretical and practical knowledge of mobile application development. • To enable students to design, implement, test, and deploy mobile applications for widely used platforms (e.g., Android, iOS). • To develop students' ability to analyze user needs and translate them into efficient, user-friendly mobile solutions. • To introduce modern tools, frameworks, and best practices in mobile app development. • To enhance teamwork, problem-solving, and innovation skills in mobile software projects.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. **Understand Flutter Fundamentals** 2. Describe the Flutter framework, its architecture, and its advantages over other cross-platform tools. 3. **Apply Dart Programming Concepts** 4. Use Dart syntax, variables, and control flow to build small console programs. 5. Implement classes, inheritance, and object-oriented principles in Flutter apps. 6. Differentiate between const, final, mutable, and immutable objects. 7. **Build and Style Flutter UIs**

	8. Use core widgets (Container, Row, Column, Stack, Text, Image, etc.) to create complex layouts. 9. Apply themes, styles, and assets effectively. 10. Create reusable widgets to improve code organization and maintainability. 11. **Implement Navigation and State Management** 12. Build multi-screen apps using MaterialApp, named routes, and dynamic navigation. 13. Use setState and widget lifecycle methods to manage UI state. 14. Pass data between screens and handle callbacks. 15. **Work with Dynamic Data** 16. Use Dart Maps and Lists to render and update data on the screen. 17. Dynamically generate widgets based on data structures. 18. **Develop a Complete Multi-Screen Application** 19. Construct a BottomNavigationBar-based app with multiple screens. 20. Integrate ticket and hotel screens, search functionality, and advanced layouts like GridView and CustomScrollView. 21. Refine UI components and polish the overall app appearance.
Indicative Contents المحتويات الإرشادية	**1. Flutter overview, app preview** 2. **Dart Fundamentals** 3. main() function, variables, and control flow 4. Classes, inheritance, and OOP concepts 5. Const, final, and immutability 6. Maps and Lists for dynamic data 7. **Core Widgets & Layout** 8. MaterialApp, Scaffold, BottomNavigationBar 9. Container, Row, Column, Stack, and basic styling 10. Images, text styling, and asset management 11. **Reusable Components & Styling** 12. Creating reusable widgets and using copyWith 13. TicketView widget and dynamic widget generation 14. AppLayoutBuilder and simple animations 15. **Navigation & Routing** 16. Named routes and dynamic navigation 17. Passing data between screens 18. **Advanced UI & Screens** 19. Hotel widget, search screen, and tabs 20. GridView and CustomScrollView 21. UI refinements and code refactoring

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	• **Lectures & Seminars**: Provide theoretical foundations and discuss emerging trends. • **Hands-on Labs**: Practical exercises using Android/iOS development environments. • **Case Studies & Demonstrations**: Analysis of real-world mobile apps and design practices. • **Project-Based Learning**: Students work individually and in teams to develop complete mobile applications. • **Problem-Solving Sessions**: Debugging, optimization, and performance challenges. • **Peer Review & Presentations**:Encourage collaborative learning, feedback, and communication skills. • Guest lectures, workshops, or hackathons to link theory with practice.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	4 , 10	2-5, 7-9
	Assignments	4	20% (20)	3,7,10,13	1, 3, 5,8,9,10
	Projects	1	20% (20)	14	18
	Report	1		9	
Summative assessment	Midterm Exam	2 hr	10% (10)		
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Introduction to Mobile Computing – Characteristics, platforms, trends& course overview
Week 2	Introduction to Flutter &Preview of the app & graphics
Week 3	Creating a Flutter project , Understanding the project structure
Week 4	Dart basics: main() function and removing comments. Classes and inheritance
Week 5	Stateful widgets and class structure & MaterialApp and Scaffold basics
Week 6	BottomNavigationBar& using setState to update navigation index
Week 7	Introduction to layout Container & Review navigation basics.
Week 8	**Midterm Exam** (Covers first half of course).
Week 9	Advanced UI Widgets : TicketView widget, Stack widget, dynamic widget generation, AppLayoutBuilderWidget, and animations
Week 10	Advanced TicketView and DART language
Week 11	Navigation and routing in depth
Week 12	Creating Hotel Porject
Week 13	Hotel Project II
Week 14	Hotel Project III
Week 15	Revision & Final Exam Preparation.

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction to visual studio code / Flutter setup.
Week 2	Creating a simple mobile app.
Week 3	DART progamming language.
Week 4	DART progamming language I
Week 5	Coding Statless & Statfull widget
Week 6	creating a BottomNavBar class, adding items and icons, styling, and dummy screens
Week 7	styling Text widgets, images, search box layout, and reusable

Week 8	Midterm Practical Test.
Week 9	TicketView widget, Stack widget, dynamic widget generation, AppLayoutBuilderWidget, and animations
Week 10	Complete TicketView: bottom parts, ternary operator, dash, reusable row widgets with Dart const, final, immutability, maps & lists, rendering and limiting data.
Week 11	routes property, dynamic navigation with callback, creating a route file
Week 12	Hotel widget creation, styling, loading map data. Search screen refactor, ticket and hotel tabs, AppTicketTabs & AppTextIcon widgets.
Week 13	FindTickets widget, plane seat layout, column stacking, Stack widget for overlapping content.TicketScreen and tabs, color customization, dividers, and reusable widgets
Week 14	Hotel detail routing.
Week 15	Final project presentation & revision.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	Beginning Flutter: A Hands On Guide to App Development by Marco L. Napoli	No
Recommended Texts	Flutter Apprentice (First Edition): Learn to Build Cross-Platform Apps	No
Websites	https://docs.flutter.dev/reference/learning-resources	

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	Operating System 1	**Module Delivery**	
Module Type	Core	☐ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	TU010103004		
ECTS Credits	8		
SWL (hr/sem)	200		
Module Level	3	**Semester of Delivery**	2st
Administering Department	Computer Science	**College**	CCSM
Module Leader	Ayad Ahmed Yass	**e-mail**	ayad.a.yass@tu.edu.iq

Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	Phd
Module Tutor		**e-mail**	
Peer Reviewer Name	Mahammed Aktham	**e-mail**	
Scientific Committee Approval Date	1/9/2025	**Version Number**	1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	Understanding basics of Operating system	**Semester**	2

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To describe the basic organization of computer systems 2. To describe the services an operating system provides to users, processes, and other systems. 3. To develop students' understanding skills for various operating system 4. To help students to deal with computer components. 5. To provide students with hands-on experience in Linux command line 6. To advance the students skills to working with scripting to deal with operating system. 7. To enhance students' ability to troubleshooting operating system problems. 8. To level up the students' skills so they be able to enter the market places since the OS consider the most required in most organizations
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Demonstrate a solid understanding of the fundamental principles and concepts of operating system. 2. to help the student to understand the components and security of the operating

	system. 3. At this point, the student will be able to deal with different operating system 4. Enhance the operating system components and avoid the security issues 5. Student able to work with Linux command as the Linux is the most required in the market nowadays 6. Less work and increase productivity by learning the scripting 7. fixing and troubleshooting all OS related issues 8. leveling up the skills means now the students can enter any market place without hesitating
Indicative Contents المحتويات الإرشادية	1- To describe the basic organization of computer systems. 2- To describe the services an operating system provides to users, processes, and other systems 3- To discuss the various ways of structuring an operating system 4- To introduce the notion of a process and a thread 5- To introduce CPU scheduling, which is the basis for multi programmed operating systems 6- To develop a description of deadlocks

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain operating concepts, like processes, threads, memory management, file systems, and system calls. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing database best practices, and exploring real-world applications of operating system 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on database exercises. Key strategies for the laboratory sessions include: a. Linux Exercises: Students will work on linux exercises and projects in the laboratory, providing them with practical experience in command line and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' skills They can help students debug their scripting, identify errors, and improve their

	command line skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on Linux tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. command line Assignments: Assignments will be given to students to reinforce their understanding of command line concepts and encourage independent problem-solving. These assignments may involve implementing scripting. 5. Reviews and Feedback: The instructor will provide feedback on students' scripting, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding command concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	93	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	107	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	7.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	#LO 1-3, #LO 5-8
	Assignments	2	10% (10)	7, 12	#LO 3-5, #LO 5-8

	Projects	1	10% (10)	continuous	
	Report	1	10% (10)	14	#LO 1-8
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	#LO 1-7
	Final Exam	2 hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction to operating system
Week 2	Computer-System Architecture Multiprocessing Architecture
Week 3	Multiprocessing Architecture
Week 4	Operating system services
Week 5	System Program
Week 6	Process 1
Week 7	Process 2
Week 8	Threads 1
Week 9	Threads 2
Week 10	CPU Scheduling 1
Week 11	CPU Scheduling 2
Week 12	Scheduling algorithms
Week 13	Scheduling Algorithms tutorials 1
Week 14	Scheduling Algorithms tutorials 2
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction about the OS to be used in the lab and why we are using it
Week 2	Differences between Linux and other operating system
Week 3	Explain about the App that using Linux

Week 4	**Installing Microsoft SQL Server Management Studio**
Week 5	Installing Linux
Week 6	How to use Linux remotely
Week 7	Linux basic command line1
Week 8	Linux command line 2
Week 9	Linux command line 3
Week 10	Installing software in Linux
Week 11	Understanding Linux logs
Week 12	Identifying issues from the logs
Week 13	Troubleshooting issues 1
Week 14	Troubleshooting issues 2
Week 15	Applying instructor's feedback on students' projects

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	*Abraham* Silberschatz (2022). Operating System Concept nine edition.	Yes
Recommended Texts		
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	**Data Science**	**Module Delivery**
Module Type	Core	☒ **Theory** ☒ **Lecture** ☒ **Lab**
Module Code	TU010103017	
ECTS Credits	5	

SWL (hr/sem)	**148**			☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Level	3		**Semester of Delivery**	first	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Yahya Layth Khaleel		**e-mail**	yahya@tu.edu.iq	
Module Leader's Acad. Title			**Module Leader's Qualification**		
Module Tutor			**e-mail**		
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date	10/9/2025		**Version Number**	1.0	

Relation with other Modules
العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents
أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Aims أهداف المادة الدراسية	• **To provide students with the fundamental concepts, theories, and methodologies of data science.** • **To enable students to collect, preprocess, analyze, and interpret data from different sources and modalities (text, images, numeric, etc.).** • **To introduce students to widely used tools, platforms, and programming environments for data science (Python, R, SQL).** • **To develop students' ability to apply statistical and machine learning algorithms for knowledge extraction.** • **To strengthen students' skills in presenting results through visualization and interpretation for effective decision-making.**

Module Learning Outcomes مخرجات التعلم للمادة الدراسية	• **Explain** the concepts, processes, and lifecycle of data science. • **Collect, clean, and prepare** data from multiple sources for analysis. • **Apply** statistical and exploratory techniques to analyze datasets. • **Use** programming tools (Python, R, Pandas, Scikit-learn, SQL) for data processing and analysis. • **Build** basic machine learning models for regression, classification, and clustering. • **Design** effective visualizations using libraries such as Matplotlib, Seaborn, or Tableau. • **Evaluate** data quality, model performance, and identify ethical issues in data use. • **Work collaboratively** on a complete data science project from collection to interpretation.
Indicative Contents المحتويات الإرشادية	• **Introduction to Data Science**: concepts, lifecycle, applications. • **Data sources and types**: structured, unstructured, big data. • **Data preprocessing**: cleaning, missing data handling, normalization, transformation. • **Exploratory Data Analysis**: descriptive statistics, distributions, correlations. • **Machine Learning Basics**: linear/logistic regression, KNN, decision trees, clustering (K-means). • **Data Management**: SQL, relational databases, integration with analysis environments. • **Data Visualization**: charts, dashboards, storytelling with data. • **Big Data Introduction**: Hadoop, Spark basics. • **Data Ethics**: privacy, bias, security. • **Practical Project**: analysis of a real dataset with reporting and presentation.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	**Lectures**: to introduce concepts, theories, and algorithms. **Laboratory Sessions**: practical training in Python, R, SQL, visualization tools. **Case Studies**: exploring real-world data applications (business, healthcare, social networks). **Group Project**: project-based learning to build a complete data science pipeline. **Seminars & Presentations**: enhancing communication of technical results. **Formative Assessments**: quizzes and exercises to reinforce continuous learning. **Industry Involvement**: guest lectures or workshops from data professionals to align with current practices.
Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا	

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)		
	Assignments	4	20% (20)		
	Projects	1	20% (20)		
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)		
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Introduction to Data Science – Definitions, lifecycle, applications.
Week 2	Data Types & Sources – Structured, unstructured, semi-structured data.
Week 3	Data Collection & Cleaning – Handling missing data, outliers.
Week 4	Exploratory Data Analysis (EDA) – Descriptive statistics, visualization.
Week 5	Probability & Statistics for Data Science – Distributions, hypothesis testing.
Week 6	Data Preprocessing – Normalization, encoding, feature selection.
Week 7	Machine Learning Basics – Supervised vs unsupervised learning.
Week 8	**Midterm Exam** (Covers first half of course).
Week 9	Regression Techniques – Linear, logistic regression.
Week 10	Classification Techniques – Decision trees, k-NN, SVM.
Week 11	Clustering & Dimensionality Reduction – k-means, PCA.
Week 12	Model Evaluation – Accuracy, precision, recall, F1, ROC curves.
Week 13	Data Visualization Tools – Matplotlib, Seaborn, Tableau/Power BI.
Week 14	Big Data & Cloud Data Science – Hadoop, Spark, cloud services.
Week 15	Ethical Issues in Data Science & Revision.

Delivery Plan (Weekly Lab. Syllabus):
المنهاج الاسبوعي للمختبر:

Week No.	Material Covered

Week 1	Introduction to Python for Data Science (Jupyter, NumPy, Pandas).
Week 2	Loading and exploring datasets (CSV, Excel, SQL).
Week 3	Data cleaning (handling missing values, duplicates, outliers).
Week 4	Exploratory Data Analysis – Visualization with Matplotlib & Seaborn.
Week 5	Probability & statistical analysis with Python (SciPy, statsmodels).
Week 6	Data preprocessing (scaling, normalization, encoding categorical data).
Week 7	Implementing simple ML models (linear regression, classification).
Week 8	**Midterm Practical Test.**
Week 9	Regression models in Python (Linear & Logistic regression).
Week 10	Classification models (Decision Trees, k-NN, SVM with scikit-learn).
Week 11	Clustering (k-means) & dimensionality reduction (PCA).
Week 12	Model evaluation metrics – Confusion matrix, ROC curve, cross-validation.
Week 13	Advanced visualization dashboards (Tableau / Power BI / Plotly).
Week 14	Introduction to Big Data tools (PySpark / Hadoop).
Week 15	Final project presentation & practical revision.

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	**Digital image processing**		**Module Delivery**	
Module Type	Core		☒ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TU010104001			
ECTS Credits	**6**			
SWL (hr/sem)	**150**			
Module Level	1	**Semester of Delivery**	2	
Administering Department	Computer science	**College**	CCSM	
Module Leader	Ahmed Saadi Abdullah Albasha	**e-mail**	ahmedalbasha@tu.edu.iq	
Module Leader's Acad. Title	Assistant Professor	**Module Leader's Qualification**	master	
Module Tutor		**e-mail**		
Peer Reviewer Name		**e-mail**		
Scientific Committee Approval Date	01/09/2025	**Version Number**	1.0	

Relation with other Modules
العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	This module aims to introduce the fundamental principles, algorithms, and applications of digital image processing. It seeks to equip students with theoretical understanding and practical skills to analyze, manipulate, and enhance images using computational methods. The module also emphasizes the role of image processing in real-world applications such as medical imaging, remote sensing, industrial inspection, multimedia, and computer vision.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Explain the fundamental concepts and mathematical foundations of digital image processing. 2. Apply spatial and frequency domain techniques for image enhancement, restoration, and analysis. 3. Implement algorithms for image filtering, segmentation, and feature extraction. 4. Evaluate the performance and limitations of different image processing methods for given tasks. 5. Use image processing software tools and programming environments (e.g., MATLAB, Python) to design and implement solutions. 6. Critically analyze case studies and research trends in image processing applications.
Indicative Contents المحتويات الإرشادية	The indicative contents of this module cover the fundamentals of digital image processing, including image formation, acquisition, sampling, and quantization. Students will study image enhancement techniques in both the spatial and frequency domains, such as histogram processing, spatial filtering, and Fourier-based filtering. The module also addresses image restoration methods for noise reduction and degradation correction, as well as color image processing using various color models and transformations. Core topics include image segmentation through thresholding, edge detection, and region-based methods, alongside morphological operations like dilation, erosion, opening, and closing for shape analysis. Feature extraction and representation methods are explored to support object recognition and analysis, complemented by image compression techniques and standards such as JPEG and PNG. The module concludes with selected applications in fields like medical imaging, remote sensing, and multimedia, reinforced by practical laboratory sessions using MATLAB or Python libraries ll types of flip-flops are studied and their work is explained in detail and how they work, as well as an explanation of the counter shift register [16 hours)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The module will be delivered through a combination of lectures, tutorials, and laboratory sessions to balance theoretical foundations with practical implementation. Lectures will introduce the key concepts, algorithms, and applications of digital image processing, while tutorials will provide guided problem-solving and discussion of case studies. Laboratory sessions using MATLAB or Python will allow students to implement and test algorithms on real datasets, reinforcing hands-on skills. Independent study and directed readings will encourage deeper exploration of advanced topics and current research. Group projects and presentations may be employed to foster collaboration, critical thinking, and communication skills.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ 15 اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	92	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	58	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	3.8
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)	5, 10	LO #1-3 , LO# 9 - 11
	Assignments	2	10% (10)	2, 12	LO # 3, 4, LO#8 -10
	Projects	1	10% (10)	Continuous	
	Report	1	10% (10)	12	LO # 5, 9 and 11
Summative assessment	**Midterm Exam**	2 hr	10% (10)	7	LO # 1-8
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction to digital image processing
Week 2	Digital image fundamentals

Week 3	Intensity transformations and histogram processing.
Week 4	Spatial domain filtering – smoothing and sharpening techniques.
Week 5	Frequency domain analysis – Fourier transform and frequency filtering.
Week 6	Image restoration – noise models, inverse and Wiener filtering.
Week 7	Color image processing – color models, transformations, and enhancement.
Week 8	Mid-term review and assessment.
Week 9	Image segmentation – thresholding methods.
Week 10	Edge detection and region-based segmentation.
Week 11	Morphological image processing – operations and applications.
Week 12	Feature extraction – texture, shape, and boundary descriptors.
Week 13	Image compression – lossless and lossy methods, standards (JPEG, PNG).
Week 14	Selected advanced topics – applications in medical imaging, remote sensing, and multimedia.
Week 15	Case studies, project discussions, and course review.
Week 16	**Preparatory week before the final exam.**

Delivery Plan (Weekly Lab. Syllabus): There is no Lab activities المنهاج الاسبوعي للمختبر:	
	Material Covered
Week 1	Introduction to MATLAB/Python environments for image processing; loading, displaying, and saving images.
Week 2	Basic image operations – grayscale conversion, resizing, cropping, intensity adjustments.
Week 3	Histogram analysis and equalization.
Week 4	Spatial filtering – implementation of smoothing and sharpening filters.
Week 5	Frequency domain filtering – applying Fourier transform, low-pass and high-pass filters.
Week 6	Noise addition and removal – experimenting with Gaussian, salt-and-pepper noise and denoising filters.
Week 7	Color image processing – color model conversions and enhancements.
Week 8	Mid-term practical assessment.
Week 9	Image segmentation – thresholding and region-growing methods.
Week 10	Edge detection techniques (Sobel, Prewitt, Canny) and comparison of results.
Week 11	Morphological operations on binary and grayscale images.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.	Yes
Recommended Texts	Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice Hall.	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	Operating System			**Module Delivery**	
Module Type	Core			☐ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	TU010104002				
ECTS Credits	8				
SWL (hr/sem)	200				
Module Level		1	**Semester of Delivery**		1st
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Huthaifa Mohammed Kanoosh		**e-mail**	huthife@tu.edu.iq	
Module Leader's Acad. Title		Lecturer	**Module Leader's Qualification**		M.Sc
Module Tutor			**e-mail**		
Peer Reviewer Name		Mahammed Aktham	**e-mail**		
Scientific Committee Approval Date		1/9/2025	**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	Understanding basics of Operating system	**Semester**	2

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To describe the advance of computer systems 2. To describe the services an operating system provides to users, processes, and other systems. 3. To develop students' understanding skills for various operating system 4. To help students to deal with computer components. 5. To provide students with hands-on experience in Linux command line 6. To advance the students skills to working with scripting to deal with operating system. 7. To enhance students' ability to troubleshooting operating system problems. 8. To level up the students' skills so they be able to enter the market places since the OS consider the most required in most organizations
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Demonstrate a solid understanding of the fundamental principles and concepts of operating system. 2. to help the student to understand the components and security of the operating system. 3. At this point, the student will be able to deal with different operating system 4. Enhance the operating system components and avoid the security issues 5. Student able to work with Linux command as the Linux is the most required in the market nowadays 6. Less work and increase productivity by learning the scripting 7. fixing and troubleshooting all OS related issues 8. leveling up the skills means now the students can enter any market place without hesitating
Indicative Contents المحتويات الإرشادية	7- To describe the basic organization of computer systems. 8- To describe the services an operating system provides to users, processes, and other systems

	9- To discuss the various ways of structuring an operating system 10- To introduce the notion of a process and a thread 11- To introduce CPU scheduling, which is the basis for multi programmed operating systems 12- To develop a description of deadlocks

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain operating concepts, like processes, threads, memory management, file systems, and system calls. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing database best practices, and exploring real-world applications of operating system 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on database exercises. Key strategies for the laboratory sessions include: a. Linux Exercises: Students will work on linux exercises and projects in the laboratory, providing them with practical experience in command line and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' skills They can help students debug their scripting, identify errors, and improve their command line skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on Linux tasks promotes discussions, problem-solving, and peer learning.

	d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. command line Assignments: Assignments will be given to students to reinforce their understanding of command line concepts and encourage independent problem-solving. These assignments may involve implementing scripting. 5. Reviews and Feedback: The instructor will provide feedback on students' scripting, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding command concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	93	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	107	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	7.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	#LO 1-3, #LO 5-8
	Assignments	2	10% (10)	7, 12	#LO 3-5, #LO 5-8
	Projects	1	10% (10)	continuous	
	Report	1	10% (10)	14	#LO 1-8
Summative	**Midterm Exam**	2 hr	10% (10)	11	#LO 1-7

assessment	**Final Exam**	2 hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Process Synchronization Part1
Week 2	Process Synchronization Part2
Week 3	Critical Section Problems part1
Week 4	Critical Section Problems part2
Week 5	Deadlock
Week 6	Method of Handling Deadlock
Week 7	Deadlock Avoidance part1
Week 8	Deadlock Avoidance part2
Week 9	Deadlock Detection part1
Week 10	Deadlock Detection part2
Week 11	Main Memory Part1
Week 12	Main Memory Part2
Week 13	Segmentation
Week 14	Segmentation and Memory Allocation
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction about the advance Linux usage
Week 2	Advance Linux Commands

Week 3	Explain about the App that using Linux
Week 4	**Introduction about the bash scripting**
Week 5	Linux Shell Scripting
Week 6	How to use shell scripting
Week 7	Advance shell scripting command
Week 8	Linux command line
Week 9	Shell Scripting Excustion
Week 10	Installing software in Linux
Week 11	Understanding Linux logs
Week 12	Identifying issues from the logs
Week 13	Shell scripting comparisons with normal command line
Week 14	Troubleshooting issues
Week 15	Applying instructor's feedback on students' projects

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	*Abraham* Silberschatz (2022). Operating System Concept Nine Edition.	Yes
Recommended Texts	Operating Systems Internals And Design Principles Seventh Edition , William Stallings	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 Security	**Module Delivery**
Module Type	Core	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**
Module Code	TU010104003	
ECTS Credits	8	
SWL (hr/sem)	200	

Module Level		1	**Semester of Delivery**		1st
Administering Department		Computer Science	**College**	CCSM	
Module Leader	**Hasan Naji Ali**		**e-mail**	hasan@tu.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		master
Module Tutor			**e-mail**		
Peer Reviewer Name		Mahammed Aktham	**e-mail**		
Scientific Committee Approval Date		1/9/2025	**Version Number**	1.0	

Relation with other Modules
العلاقة مع المواد الدراسية الأخرى

Prerequisite module	None	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents
أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Aims أهداف المادة الدراسية	• Introduce the fundamental principles and goals of computer and information security. • Explain the main security concepts: confidentiality, integrity, availability, authentication, and non-repudiation. • Present the major types of security threats and attacks (malware, phishing, social engineering, etc.). • Familiarize students with cryptographic concepts and access control mechanisms.

	• Enable students to understand authentication methods, digital certificates, and secure communication protocols. • Develop practical skills in securing computer systems, operating systems, and networks. • Raise awareness about ethical, legal, and professional issues in cybersecurity. • Prepare students for future work or research in information security and cyber defense.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	• Upon successful completion of this module, students should be able to: • Describe the principles and objectives of computer security. • Identify and analyze security threats, vulnerabilities, and mitigation techniques. • Apply access control and authentication mechanisms in practical scenarios. • Understand the role of cryptography in data protection and communication security. • Implement basic secure configurations in operating systems and network environments. • Evaluate real-world attack vectors and suggest appropriate countermeasures. • Demonstrate ethical awareness and adherence to cybersecurity best practices.

Indicative Contents
المحتويات الإرشادية
	• **Introduction to Computer Security**
	
	  • CIA Triad (Confidentiality, Integrity, Availability)
	  • Security goals and key principles
	  • Threats, vulnerabilities, and risks
	
	• **Types of Attacks and Malware**
	
	  • Viruses, worms, Trojans, ransomware
	  • Social engineering and phishing attacks
	  • Insider and outsider threats

	• **Authentication and Access Control**
	  • Identification, authentication, and authorization
	  • Passwords, OTP, biometrics, and MFA
	  • Access control models (DAC, MAC, RBAC)
	• **Cryptography Basics**
	  • Symmetric and asymmetric encryption
	  • Hash functions and digital signatures
	  • Public Key Infrastructure (PKI)
	• **Network and Web Security**
	  • Firewalls, VPNs, and intrusion detection systems
	  • Secure communication protocols (SSL/TLS, HTTPS)
	  • Wi-Fi and web application security
	• **Operating System Security**
	  • User accounts, permissions, and auditing
	  • Patch management and system hardening
	• **Incident Response and Risk Management**
	  • Risk assessment and mitigation
	  • Backup, recovery, and business continuity
	• **Cyber Ethics and Legal Issues**
	  • Privacy, data protection, and intellectual property
	  • Ethical hacking and professional conduct

Learning and Teaching Strategies

استراتيجيات التعلم والتعليم

Strategies	• **Lectures:** Theoretical explanation of core concepts and real-world case studies. • **Labs:** Hands-on exercises using security tools (e.g., Wireshark, Nmap, HashCalc). • **Assignments:** Security configuration and report writing. • **Quizzes & Discussions:** Reinforcing understanding of key topics. • **Project:** Small team-based security analysis or penetration testing simulation.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	60	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	4
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	140	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	9.4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Due Week	Relevant Learning Outcome
Formative assessment	**Quizzes**	3	15% (15)	5, 7, 11	
	In-Class Activity	1	5% (5)	9	
	Assignments	2	10% (10)	6, 12	
	Projects	1	10% (10)	5-14	
	Report				
Summative assessment	**Midterm Exam**	1 hr.	10% (10)	11	
	Final Exam	2 hr.	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction to Computer Security and CIA Triad
Week 2	Threats, Vulnerabilities, and Attacks
Week 3	Malware Types and Social Engineering
Week 4	Authentication and Access Control
Week 5	Passwords, Biometrics, and MFA
Week 6	Cryptography Fundamentals
Week 7	Symmetric and Asymmetric Encryption
Week 8	Midterm Exam
Week 9	Firewalls and IDS/IPS
Week 10	Operating System Security
Week 11	Risk Management and Incident Response
Week 12	Cyber Ethics and Legal Frameworks
Week 13	Security Tools Demonstration
Week 14	Project Presentations and Review
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Lab orientation and security awareness basics
Week 2	Configuring user accounts and permissions
Week 3	Implementing password policies
Week 4	Encryption and hashing experiments
Week 5	Identifying threats and malware samples (simulation)
Week 6	Packet analysis using Wireshark

Week 7	Simulated phishing attack and defense
Week 8	VPN and secure connection setup
Week 9	Log analysis and system auditing
Week 10	Intrusion detection (Snort or Suricata demo)
Week 11	Backup and disaster recovery exercise
Week 12	Ethical hacking lab practice
Week 13	Security report writing
Week 14	Fixing problems in students' projects
Week 15	Final project demonstration

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	➢ Computer Security, Principle and Practice, Fourth Edition	(Soft copy)
Recommended Texts	➢ Effective Cybersecurity, William Stallings ➢ CRYPTOGRAPHY AND NETWORK SECURITY, Principles and Practice, WILLIAM STALLINGS, EIGHTH EDITION	(Soft copy) (Soft copy)
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 Network			Module Delivery	
Module Type	Core			☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☒ Seminar	
Module Code	TU010104004				
ECTS Credits					
SWL (hr/sem)					
Module Level	4		Semester of Delivery	first	
Administering Department	Computer Science		College	CCSM	
Module Leader	Majid Hamid Ali		e-mail	Majid.hamid@tu.edu.iq	
Module Leader's Acad. Title			Module Leader's Qualification		
Module Tutor	Kahlan		e-mail		
Peer Reviewer Name			e-mail		
Scientific Committee Approval Date	13/01/2025		Version Number	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	• Teaching the student the theoretical skills of data communications and networks • Qualifying students to teach this subject to middle and middle school students • Preparing students to work in the private and governmental telecommunications sector • Teaching the student theoretical computer networking skills • Understanding computer networks. • Understanding the layers of computer networks. • Understanding the protocols and the packet's journey from source to target. • Knowledge of basic concepts of computer networks • The student recognizes the importance of general concepts of computer networks. • Identify the most important protocols used to organize the network's work • Study the most important factors that affect the work of networks
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Explain the fundamental concepts, models, and protocols of computer networks, including the OSI and TCP/IP architectures. 2. Describe key networking components such as routers, switches, and transmission media, and their roles in data communication. 3. Analyze how data is transmitted, routed, and controlled across

	different layers of a network. 4. Demonstrate practical skills in configuring and troubleshooting basic network setups using simulation tools (e.g., Cisco Packet Tracer, GNS3). 5. Compare and evaluate different networking technologies, including wired and wireless networks, LANs, WANs, and the Internet. 6. Apply network addressing schemes (such as IPv4 and IPv6) and subnetting in designing efficient network architectures. 7. Assess issues related to network performance, security, and reliability, and propose suitable solutions. 8. Collaborate effectively in teams to plan, design, and implement small-scale network projects.

Indicative Contents
المحتويات الإرشادية
	• **Introduction to Computer Networks**
	  • Definition, goals, and importance of networking
	  • Network types: LAN, MAN, WAN, WLAN, PAN
	  • Network topologies and their characteristics
	  • Network hardware and software components
	• **Network Models and Architectures**
	  • OSI Reference Model: layers, functions, and protocols
	  • TCP/IP Protocol Suite and comparison with OSI model
	  • Encapsulation and data flow through layers
	• **Physical Layer Concepts**
	  • Data and signals: analog vs. digital
	  • Transmission media: guided (copper, fiber optics) and unguided (wireless)
	  • Bandwidth, throughput, and data rate
	  • Signal encoding, multiplexing, and switching techniques
	• **Data Link Layer**
	  • Framing, error detection, and correction
	  • Flow control and access control (CSMA/CD, CSMA/CA)
	  • Ethernet standards (IEEE 802.3) and LAN technologies
	  • Point-to-Point Protocol (PPP) and HDLC
	• **Network Layer**
	  • IP addressing (IPv4, IPv6) and subnetting
	  • Routing concepts and algorithms (RIP, OSPF, BGP)
	  • Switching and routing devices
	  • ICMP and ARP protocols

	• **Transport Layer**
	  • Connection-oriented and connectionless services
	  • TCP and UDP protocols
	  • Port numbers, sockets, and multiplexing
	  • Flow control and congestion control
	• **Application Layer**
	  • Domain Name System (DNS)
	  • Email (SMTP, POP3, IMAP)
	  • Web services (HTTP, HTTPS)
	  • File transfer (FTP, TFTP)
	• **Network Security and Management**
	  • Network threats and vulnerabilities
	  • Firewalls, encryption, and VPNs
	  • Network monitoring and management tools (SNMP)
	  • Introduction to cybersecurity concepts
	• **Wireless and Mobile Networking**
	  • Wi-Fi standards (IEEE 802.11)
	  • Cellular networks and mobile IP
	  • Ad hoc and sensor networks
	• **Emerging Networking Technologies**
	  • Software Defined Networking (SDN)
	  • Internet of Things (IoT) basics
	  • Cloud networking and data centers
	• **Practical and Laboratory Work**
	  • Network configuration using simulation tools (Packet Tracer, GNS3)
	  • IP addressing and subnetting exercises
	  • Designing and troubleshooting small networks
	  • Performance testing and analysis

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	

Strategies	1. **Lectures**
	  o Used to explain theoretical foundations, key principles, and network architectures.
	  o Supported by visual aids such as diagrams, animations, and case studies.
	2. **Laboratory Sessions / Practical Work**
	  o Students apply theoretical knowledge using simulation tools such as **Cisco Packet Tracer**, **GNS3**, or real networking devices.
	  o Exercises include IP addressing, subnetting, routing configuration, and troubleshooting network problems.
	3. **Tutorials and Discussion Sessions**

	○ Small-group discussions to clarify difficult topics and encourage critical thinking. ○ Problem-solving exercises to develop analytical and design skills. 4. **Project-Based Learning** ○ Students work in teams to design and implement a small-scale network or simulation project. ○ Encourages collaboration, planning, and application of theoretical knowledge in real-world contexts. 5. **E-Learning and Digital Resources** ○ Use of learning management systems (LMS) for sharing lecture notes, assignments, and supplementary materials. ○ Integration of online videos, network simulators, and interactive quizzes to enhance learning. 6. **Case Studies and Industry Examples** ○ Presentation of real-world networking scenarios from local or international organizations. ○ Encourages understanding of current technologies and industry standards. 7. **Self-Directed Learning** ○ Students are encouraged to explore advanced topics independently. ○ Reading assignments, online tutorials, and research activities to deepen understanding.

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

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	Structured SWL (h/w) الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	Unstructured SWL (h/w) الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	4 , 10	2-5, 7-9
	Assignments	4	20% (20)	3,7,10,13	1, 3, 5,8,9,10
	Projects	1	20% (20)	14	18
	Report	1		9	
Summative assessment	Midterm Exam	2 hr	10% (10)		
	Final Exam	2hr	40% (40)	16	All

Total assessment	100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	Material Covered
Week 1	Definition: Packet Tracer
Week 2	Identify the physical connection of devices
Week 3	To introduce the means of transmission, wired and wireless, and how to link them in the program
Week 4	Learn about ways to send the signal
Week 5	Identify the logical connection of devices
Week 6	Types of devices in networks
Week 7	Networking software
Week 8	Introduction to the OSI model
Week 9	Introduction to the TCP/IP protocol
Week 10	Identify network protocols
Week 11	Learn about Switching Techniques
Week 12	Learn how to connect different networks
Week 13	Learn about data transfer
Week 14	How to address within the network
Week 15	Learn about the Internet and its applications

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction to Packet Tracer and Network Devices -
Week 2	Introduction to Switches & Packet Tracer
Week 3	Basic Switch Configuration
Week 4	VLAN Concepts
Week 5	Inter-VLAN Routing Basics
Week 6	VLAN Trunking Protocol (VTP)

Week 7	Trunk Links
Week 8	Port Security
Week 9	Spanning Tree Protocol (STP) Basics
Week 10	Advanced STP (RSTP)
Week 11	EtherChannel
Week 12	Switch Troubleshooting Basics
Week 13	DHCP Snooping
Week 14	Dynamic ARP Inspection (DAI)
Week 15	Private VLANs

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	Available in the Library?
Required Texts	1."TCP/ IP Protocol Suites", Behrouz Forouzan, McGraw-Hill, 4th edition, 2010	No
Recommended Texts	2. "Data Communication And Networking", Behrouz a. Forouzan, 4th edition, 2009	No
Websites	http://www.youtube.com/playlist?list=PL828D58CF32F123B6	

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	Machine Learning			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	TU010104005				
ECTS Credits	4				
SWL (hr/sem)	148				
Module Level		2	**Semester of Delivery**		4th
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Armaneesa Naaman Hasoon		**e-mail**	armaneesa@tu.edu.iq	
Module Leader's Acad. Title		Assistant Professor	**Module Leader's Qualification**		ماجستير
Module Tutor			**e-mail**		
Peer Reviewer Name		Dr. Mohammed Aktham	**e-mail**		
Scientific Committee Approval Date		1/09/2025	**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Artificial intelligence	**Semester**	
Co-requisites module	Python Calculus, Probability	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية

Module Aims أهداف المادة الدراسية	•Understand the term machine learning and its applications •Understand the steps of machine learning. •Study the types of learning •Solve problems that traditional methods cannot solve. •Understand the methods of obtaining and processing data •Understand the methods of extracting features • Train and evaluate machine learning models
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	•Understand the definition and applications of machine learning •Understand data processing methods. •Develop the student's thinking skills and the ability to build a machine learning model used for prediction or classification •Provide the learner with the skills necessary to be able to evaluate the performance of the machine learning model and apply it in real-time
Indicative Contents المحتويات الإرشادية	• Machine Learning Basics •Problem in AI (Problem State Space, Search Space, Problem Solution) •Search Classification in AI •Blind Search Algorithms •Heuristic Search Algorithms •Machine Learning Basics

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1 .Lectures: The instructor will give lectures to introduce and explain the definition of machine learning methods and their applications. 2 .Interactive discussions: Engaging students in interactive discussions allows them to ask questions, seek clarification, and actively participate in learning. Discussions can include examples of machine learning algorithms, discuss how these algorithms are programmed in practice, and explore real-world applications of AI concepts. 3 .Lab: Lab sessions are dedicated, hands-on sessions where students apply the concepts they have learned in lectures to practical programming exercises. Key strategies for lab sessions include: a. Programming exercises: Students will work on programming exercises and projects in the lab, providing them with hands-on experience in programming machine learning

	projects. b. Guided training: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and peer learning: Students can collaborate with their peers in the lab, promoting teamwork and enabling knowledge sharing. Working together on machine learning project programming assignments promotes discussion, problem solving, and peer learning. D. Access to Equipment and Resources: The lab should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the resources needed to complete lab exercises and assignments effectively. 4 .Programming Assignments: Students will be given assignments to enhance their understanding of how to program machine learning models to encourage independent problem solving. These assignments may include implementing algorithms, designing software systems, or developing small-scale projects using Python. 5 .Code Reviews and Feedback: The instructor will provide feedback on students' code, review their solutions, and make suggestions for improvement. This feedback will help students improve their coding skills and adhere to best practices. 6 .Office Hours and Individual Support: The instructor should provide individual consultations and support for students who need additional help or guidance in understanding and completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

	Time/Nu	Weight (Marks)	Week Due	Relevant Learning

		mber			Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	9-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction to machine learning:
Week 2	Importance of Machine Learning and its Applications
Week 3	Types of Machine Learning techniques and the Differences Between Supervised and Unsupervised Learning
Week 4	Steps of a Machine Learning Model in Detail
Week 5	The performance evaluation metrics for machine learning methods
Week 6	Supervised Learning:
Week 7	Classification: Introduction to Classification, Decision tree, Support Vector Machines, Random Forest
Week 8	Regression: Introduction to Regression, Simple Linear Regression, Multi-linear regression.
Week 9	Mid Exam
Week 10	Unsupervised Learning:
Week 11	Clustering: Introduction to Clustering, K-Means, Hierarchical clustering
Week 12	Dimensionality reduction
Week 13	Reinforcement learning
Week 14	Project Due (Students Presentations part1)
Week 15	Project Due (Students Presentations part2)

Delivery Plan (Weekly Lab. Syllabus):

المنهاج الاسبوعي للمختبر:

Week No.	Material Covered
Week 1	Introduction to the basics of Python programming Language
Week 2	Use libraries like Pandas, Numpy, and Matplotlib to analyze datasets.
Week 3	Build a Simple Linear Regression using Scikit-learn.
Week 4	Build a multiple variables Regression using Scikit-learn.
Week 5	Build a Logistic Linear Regression using Scikit-learn.
Week 6	Build a Decision tree classifier using Scikit-learn.
Week 7	Build a Decision tree classifier using Scikit-learn.
Week 8	Build a Random Forest classifier using Scikit-learn.
Week 9	Build a Support vector machine classifier using Scikit-learn.
Week 10	Build a K-Means clustering method using Scikit-learn.
Week 11	Build a hierarchical clustering using Scikit-learn.
Week 12	Build a PCA using Scikit-learn.
Week 13	Build a PCA using Scikit-learn.
Week 14	Answering students' questions and reviewing previous topics.
Week 15	Answering students' questions and reviewing previous topics.

Learning and Teaching Resources

مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition by Aurélien Géron (Author), 2017.	No
Recommended Texts		
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	Advanced Programming			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code					
ECTS Credits	6				
SWL (hr/sem)	148				
Module Level		1	**Semester of Delivery**		2nd
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Mohanad Hatem Ramadhan		**e-mail**	Mohanad.H.Ramadhan@tu.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		M.Sc.
Module Tutor			**e-mail**		
Peer Reviewer Name		Mohamed Aktham	**e-mail**		
Scientific Committee Approval Date		1/02/2026	**Version Number**		1.0

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. Understanding Advanced Data Structures: The module aims to provide students with a deep understanding of arrays, strings, and their manipulation techniques. Students will learn about multidimensional arrays, character arrays, and string handling functions. 2. Mastery of Pointers: The module aims to develop students' proficiency in using pointers in C++. Students will learn the concepts of memory addresses, pointer arithmetic, and dynamic memory allocation. They will understand how to manipulate data using pointers and how to utilize them for efficient memory management. 3. File Handling and Input/Output Operations: The module aims to introduce students to file handling concepts and techniques in C++. Students will learn how to read from and write to files, open and close files, handle file errors, and perform various input/output operations using file streams. They will understand file modes, buffering, and error handling. 4. File Management and Organization: The module aims to teach students how to manage and organize files effectively in C++. They will learn to create, modify, and delete files, organize file directories, and handle file-related operations. Students will understand the importance of file management in real-world programming scenarios. 5. Practical Application and Problem-Solving: Throughout the module, students will be exposed to practical programming exercises and problem-solving tasks. They will apply the concepts learned to solve real-world programming challenges, consolidating their understanding and enhancing their problem-solving skills. By focusing on arrays, strings, pointers, and file handling in C++, this advanced programming module aims to provide students with a comprehensive understanding of these concepts and their practical application. Students will develop the skills necessary to manipulate complex data structures, handle files, and write efficient and reliable code.

Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Demonstrate an in-depth understanding of arrays, strings, pointers, and file handling concepts in C++. 2. Apply advanced array operations, such as searching and sorting algorithms, and multidimensional arrays to solve programming problems. 3. Manipulate strings effectively, including concatenation, substring extraction, searching, and sorting. 4. Utilize pointers proficiently for data manipulation, including memory addresses, and pointer arithmetic 5. Read from and write to files, perform input/output operations, and handle file-related errors using file streams in C++. 6. Manage and organize files effectively, including creating, modifying, deleting, and organizing file directories. 7. Apply efficient programming techniques, optimize code, and adhere to best practices for writing clean and readable code. 8. Demonstrate problem-solving skills by applying the learned concepts to solve real-world programming challenges. 9. Work collaboratively in teams, communicate effectively, and share knowledge and ideas related to advanced programming concepts. 10. Adapt to new programming concepts and technologies beyond the scope of the course, building a foundation for lifelong learning in programming. These learning outcomes reflect the knowledge, skills, and competencies that students will acquire upon completing the advanced programming course. The outcomes emphasize both theoretical understanding and practical application, preparing students for real-world programming challenges and further studies in the field of computer science.
Indicative Contents المحتويات الإرشادية	1. Review of Basic Programming Concepts: - Recap of fundamental programming concepts, including variables, data types, control structures, and functions. 2. Arrays: - Multidimensional arrays - Array manipulation techniques - Searching and sorting algorithms 3. Strings: - String manipulation and operations - String handling functions 4. Pointers: - Introduction to pointers and their usage - Memory addresses and pointer arithmetic - Pointers to arrays

	5. Files: - File handling concepts - Reading from and writing to files - File organization and management

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain programming concepts, C++ syntax, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning.

	d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of programming concepts and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using C++. 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	5-14	

	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Recap of fundamental programming concepts, including variables, data types, control structures, and functions.
Week 2	Introduction to Arrays (Linear arrays)
Week 3	Searching and Sorting Linear Arrays
Week 4	Multidimensional Arrays and Square Arrays
Week 5	Multiplication of Two Arrays and Re-write TicTacToe game with Arrays
Week 6	Introduction to String and Its Operations
Week 7	More Examples on String
Week 8	Introduction to Pointers
Week 9	Pointer to Array and Pointer Arthmetic
Week 10	First Project Due (Reviewing and Comments)
Week 11	Introduction to Files and Directories
Week 12	Working with Text Files (Read, Write)
Week 13	Working with Binary Files
Week 14	Second Project Due (Students Presentations part1)
Week 15	Second Project Due (Students Presentations part1)

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered

Week 1	Getting used to CLI Interfaces and practicing some commands on PowerShell
Week 2	Running Examples on Array
Week 3	Practicing Arrays further (Searching)
Week 4	Practicing Arrays further (Sorting)
Week 5	Running Examples on 2D and Square Arrays
Week 6	Running Examples on Strings
Week 7	Searching in String
Week 8	Running Characters Frequency Example
Week 9	Running Examples on Pointers
Week 10	Running More Examples on Pointers
Week 11	Running Examples on Directories and Files
Week 12	Running More Examples on Files
Week 13	Running More Advanced Programs on Files
Week 14	Wrapping up
Week 15	Answering Students Questions and Extra Advising on Real World Application Programming

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?

Required Texts	Stroustrup, Bjarne - Programming_ principles and practice using C++-Addison-Wesley (2015)	Yes
Recommended Texts	Olsson, Mikael - C++20 Quick syntax reference: a pocket guide to the language, apis, and library	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	Algorithm Design			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	4				
SWL (hr/sem)	108				
Module Level		1	**Semester of Delivery**		2nd
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Firas Layth		**e-mail**		
Module Leader's Acad. Title		Assistant Prof.	**Module Leader's Qualification**		PhD
Module Tutor			**e-mail**		
Peer Reviewer Name		Mahammed Aktham	**e-mail**		
Scientific Committee Approval Date		01/02/2026	**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	The aim of this module is to introduce students to the fundamental concepts of algorithms, algorithm design, and problem-solving techniques. The module will cover various algorithmic paradigms, data structures, and analysis methods to equip students with the skills necessary for designing and analyzing algorithms effectively.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Understand the importance of algorithms in computer science and the significance of algorithmic problem-solving. 2. Design algorithms using flowcharts and pseudocode, and implement them using programming constructs such as flow control statements and loops. 3. Analyze the time and space complexity of algorithms using Big O notation and asymptotic analysis. 4. Implement and utilize basic data structures such as arrays, strings, stacks, and queues for algorithmic problem-solving. 5. Apply various sorting and searching algorithms, including bubble sort, selection sort, insertion sort, quicksort, mergesort, heapsort, linear search, binary search, depth-first search, and breadth-first search. 6. Utilize string algorithms for pattern matching and string manipulation tasks. 7. Demonstrate the ability to review and evaluate projects related to algorithm design and implementation.
Indicative Contents المحتويات الإرشادية	1. Introduction to algorithms: Understanding the role and significance of algorithms in computer science. 2. Algorithmic problem-solving: Exploring strategies and techniques for solving computational problems effectively. 3. Algorithm design: Drawing flowcharts and writing pseudocode to represent algorithmic solutions. 4. Flow control: Implementing flow control statements (if-else, switch-case) for decision-making in algorithms. 5. Loops: Utilizing loops for repetitive tasks, including counter and cumulative variables, and nested loops. 6. Complexity analysis: Analyzing the time and space complexity of algorithms using Big O notation and asymptotic analysis. 7. Basic data structures: Introduction to arrays, strings, stacks, and queues for storing and manipulating data. 8. Sorting algorithms: Implementing and analyzing sorting algorithms such as bubble sort, selection sort, insertion sort, quicksort, mergesort, and heapsort. 9. Searching algorithms: Implementing and analyzing searching algorithms such as linear search, binary search, depth-first search, and breadth-first search.

	10. String algorithms: Exploring algorithms for pattern matching and string manipulation tasks. 11. Reviewing students' projects: Providing feedback and evaluation on projects related to algorithm design and implementation.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	Lectures: Traditional lectures can be used to introduce key concepts, theories, and principles related to algorithms. Lectures should be interactive, incorporating examples, demonstrations, and real-world applications to illustrate abstract concepts effectively. Group Discussions: Group discussions encourage collaborative learning and critical thinking. Students can discuss challenging topics, share insights, and work together to solve algorithmic problems. Group discussions also promote communication skills and teamwork. Problem-Solving Sessions: Dedicated problem-solving sessions allow students to practice applying algorithmic techniques to solve a variety of problems. These sessions can involve solving algorithmic puzzles, coding challenges, and algorithm design exercises individually or in groups. Practical Coding Assignments: Assigning practical coding assignments allows students to implement algorithms and data structures in programming languages of their choice. Through coding assignments, students gain hands-on experience with algorithm implementation, debugging, and optimization. Case Studies: Case studies provide real-world examples of how algorithms are used to solve practical problems in various domains, such as finance, healthcare, and engineering. Analyzing case studies helps students understand the relevance and applicability of algorithms in different contexts.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	52	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	3.4

Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	108

Module Evaluation
تقييم المادة الدراسية

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

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Importance of algorithms in computer science
Week 2	Importance of algorithmic problem-solving
Week 3	Algorithms Design Drawing Flowchart and Writing pseudocode
Week 4	Flow Control (if-else), (switch – case)
Week 5	Loops (counter and cumulative variables), Nested Loops
Week 6	Time complexity analysis (Big O notation), Space complexity analysis and Asymptotic analysis
Week 7	Midterm exam
Week 8	Basic Data Structures: Arrays, Strings, Stacks, Queues.
Week 9	Sorting Algorithms: Bubble sort, selection sort, insertion sort
Week 10	Sorting Algorithms: Quicksort, mergesort, heapsort
Week 11	Searching Algorithms: Linear search, binary search
Week 12	Searching Algorithms: Depth-first search, breadth-first search

Week 13	String Algorithms: Pattern matching algorithms
Week 14	String Algorithms: String manipulation techniques
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction to Algorithm Design Overview of the course objectives and expectations Introduction to algorithm design methodologies Hands-on activity: Drawing flowcharts for simple algorithms Assignment: Practice drawing flowcharts for algorithmic problems
Week 2	Review of pseudocode and its importance in algorithm design Introduction to flow control statements (if-else, switch-case) Hands-on activity: Writing pseudocode for algorithmic problems Assignment: Implementing algorithms using flow control in a programming language
Week 3	Understanding loop structures and their importance in algorithms Hands-on activity: Implementing loops for counter and cumulative variables Introduction to nested loops Assignment: Solving algorithmic problems using nested loops

Week 4	Time Complexity Analysis Introduction to time complexity analysis using Big O notation Understanding the concept of asymptotic analysis Hands-on activity: Analyzing the time complexity of algorithms Assignment: Analyzing the time complexity of sorting algorithms
Week 5	Space Complexity Analysis Introduction to space complexity analysis Hands-on activity: Analyzing the space complexity of algorithms Assignment: Analyzing the space complexity of searching algorithms
Week 6	Basic Data Structures Introduction to arrays, strings, stacks, and queues Hands-on activity: Implementing basic data structures in a programming language Assignment: Implementing algorithms using basic data structures
Week 7	Sorting Algorithms Introduction to sorting algorithms: bubble sort, selection sort, insertion sort Hands-on activity: Implementing sorting algorithms Assignment: Comparing the performance of different sorting algorithms
Week 8	Sorting Algorithms (continued) Introduction to more advanced sorting algorithms: quicksort, mergesort, heapsort Hands-on activity: Implementing advanced sorting algorithms Assignment: Optimizing sorting algorithms for different datasets

Week 9	Searching Algorithms Introduction to searching algorithms: linear search, binary search Hands-on activity: Implementing searching algorithms Assignment: Analyzing the performance of searching algorithms
Week 10	Graph Algorithms Introduction to graph algorithms: depth-first search, breadth-first search Hands-on activity: Implementing graph traversal algorithms Assignment: Solving graph-related problems using depth-first search and breadth-first search
Week 11	String Algorithms Introduction to string matching algorithms Hands-on activity: Implementing pattern matching algorithms Assignment: Applying string manipulation techniques to solve algorithmic problems
Week 12	Review and Project Work
Week 13	Project Work and Consultation Project work: Students continue working on their projects Individual consultations with the instructor for project guidance and feedback

Week 14	Project Presentation Preparation Preparation for project presentations Practice sessions for project presentations Final touches on project implementations and documentation
Week 15	Project Presentations

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Introduction to Algorithms, Third Edition By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein	No
Recommended Texts	Introduction to Algorithmic Design and Analysis	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	Computation theory			**Module Delivery**	
Module Type	Supportive			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	6				
SWL (hr/sem)	150				
Module Level	1		**Semester of Delivery**	2	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Rasha Abdullah yousif		**e-mail**	Rasha.a.yousif@tu.edu.iq	
Module Leader's Acad. Title	Assistant Lecturer		**Module Leader's Qualification**	master	
Module Tutor			**e-mail**		
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date	01/02/2026		**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	Theory of computation is the theoretical study of capabilities and limitations of Computers (Theoretical models of computation). Providing students with: 1. Understand 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. 4. Be familiar with standard tools and notation for formal reasoning about machines and programs.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: Cognitive objectives: 1. Enabling the student to know and understand the theoretical principles of the programming process. 2. Enabling the student to know the intellectual framework of computer basics. 3. Enabling the student to know the practical applications of the programming foundation that the subject has. 4. Enabling the student to explain the steps of program implementation. kills objectives for the course: 1. Designing basic diagrams for computational theoretical methods. 2. Writing algorithms. 3. Writing and implementing methods. 4. Following up on the implementation and planning of algorithms and discovering errors. Teaching and learning methods, theoretical lectures, using projectors for clarification with illustrative examples, drawing diagrams that facilitate the process of understanding and conveying the idea to the student.
Indicative Contents المحتويات الإرشادية	Focuses interest on studying how calculations and reasoning are performed in the context of computer systems. 1. Computing Paradigms: Paradigms such as the Turing machine, quantum computing are explored and used to analyze computational capabilities and limits.

	2. Description Languages: Computational theory studies the languages and models that are used to describe and analyze computational operations, such as neural language and conditional language. 3. Computational complexity: It is concerned with analyzing computer complexity and classifying problems according to their computational difficulty and the ability of computer systems to solve them. 4. Mental Machines: Computational theory studies the relationship between computing and the human mind, and whether the mind can be represented and simulated by computational machines. 5. Artificial Intelligence: Artificial intelligence is an important application of computational theory, where computational models are used to develop artificial intelligence systems capable of thinking and making decisions.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain programming concepts, C++ syntax, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of programming concepts and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using C++. 6. Office Hours and Individual Support: The instructor should be available for

	individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL)
الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	77	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	5.1
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	73	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	4.8
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)	5, 11	LO #1-4 , LO# 5-7
	Assignments	2	10% (10)	7, 12	LO #1-4 , LO# 5-7
	Projects	1	10% (10)	continuous	LO #1-7
	Report	1	10% (10)	14	LO # 4 - 7
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	LO # 1-6
	Final Exam	2 hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Set, string, alphabet and language
Week 2	Regular expression
Week 3	Finite state automata, deterministic and nondeterministic finite state automata.
Week 4	Equivalence between deterministic and nondeterministic finite state automata

Week 5	Finite state automata with empty move.
Week 6	Equivalence of NFA with and without ϵ move
Week 7	The equivalence between Moore and Mealy machine
Week 8	Grammar, Chomsky hierarchy of languages.
Week 9	- The regular grammars and regular languages.
Week 10	- Closure properties of regular sets (union, concatenation and kleen closure).
Week 11	Regular expression, closure properties of regular languages (intersection, complementation and substitution)
Week 12	Decision procedures for regular sets(emptiness, finiteness, containment and equivalence).
Week 13	Context- free grammars and languages with their Properties.
Week 14	Context-free grammar without empty string production (λ-free grammar).
Week 15	Derivation trees.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Introduction to Computer Theory 3 rd Edition michael sipser, USA, 2013. ISBN-13: 978-1-133-18779-0.	Yes
Recommended Texts	Introduction to Automata Theory, Languages, and Computation, 2/E, John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, Addison-Wesley 2001. ISBN 0-201-44124-1.	No
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					
ECTS Credits	6				
SWL (hr/sem)	148				
Module Level		1	**Semester of Delivery**		2
Administering Department		Computer science	**College**	CCSM	
Module Leader	Nayif M. Hamash		**e-mail**		
Module Leader's Acad. Title		Assistant Professor	**Module Leader's Qualification**		master
Module Tutor			**e-mail**		
Peer Reviewer Name		Ahmed Saadi Abdullah	**e-mail**		
Scientific Committee Approval Date		01/02/2026	**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To develop problem solving skills and understanding of logic design 2. Teaching students the computer numerical systems 3. Teaching the student the logic gates, their structure, and the truth table 4. Teaching the student the sequential logic circuits, their analysis and installation 5. Teaching students to design and analyze counters and registers and their types 6. Teaching the student the flip flop, its types, the state table and the state diagram
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Knowledge of logic circuits and their design and analysis of complex and synchronous 2. Install logical gates and configure logical circuits 3. Application of logical circuits and their use in random memory 4. Mastering the work of gates and logical circuits and methods of connecting them 5. Knowing ways to connect and deal with half Adder as well as full Adder and how to connect their logical circuits 6. Know how to connect and deal with half Subtract as well as full Subtract and how to connect their logical circuits 7. Know the logical structure of multiplexer and how to connect more than one multiplexer in one logical circuit 8. Knowledge of the logical structure of Flip_Flop of all kinds and how to connect more than one Flip_Flop in one logical circuit
Indicative Contents المحتويات الإرشادية	Indicative content includes the following: <u>Part A – Fundamental concepts</u> Numerical systems and how to convert between them and algebraic and logical operations on them, study logical gates and study how to draw circles and form logical circles from these gates and how to write the expression through drawing [12 hours] <u>Part B- Logic Expressions Simplify, Adder, Subtract</u> The laws of Boolean algebra are studied to simplify Boolean expressions. The K_map method is also studied and applied to simplify Boolean expressions. The addition process is studied based on half Adder and full Adder, in addition to learning the subtraction process based on half subtract and full subtract. [16 hours]

	Part C- multiplexer ,Decoder A multiplexer, Demultiplaxer and how to connect the function of each of them are studied, in addition to how to form multiplexers of large sizes from multiplexers of small sizes. Also, the decoder, encoder and how to connect their logical circuits are understood in detail, in addition to how to connect them together in one circuit. [16 hours] Part D- Flip_flop types ll types of flip-flops are studied and their work is explained in detail and how they work, as well as an explanation of the counter shift register [16 hours)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	This course is characterized by the fact that it needs a special approach based mainly on the development of engineering thinking and the mathematical approach to thinking. K. Teaching depends mainly on homework that is presented at the end of each week, and the student notes the interdependence between the topics of the series in this course, in addition to assigning the student (or a group of students) to write one report and present it as a seminar for the purpose of training in the use of scientific resources, in addition to Assigning the student with a set of practical experiments that he implements in the laboratory and others that are given as homework, which will help the student to understand more broadly how logical circuits work

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation
تقييم المادة الدراسية

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

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	Introduction to logic Design and numerical systems
Week 2	Convert between numerical system and mathematical operation on each system
Week 3	Logical gates and how to draw them, and draw logic expression
Week 4	Simplifying Boolean expressions based on the laws of Boolean algebra
Week 5	Simplify Boolean expressions based on K_map (sop) ,(pos)
Week 6	Using full Adder and half Adder in addition
Week 7	Use full subtract and half subtract in subtraction
Week 8	Learn how multiplexers and demultiplexers work and how to connect their logic circuits
Week 9	Learn the work of the decoder and encoder and how to connect their logic circuits
Week 10	Learn about the working of J-K flip flop
Week 11	Learn about the working of J-K flip flop
Week 12	Learn about the working of T flip flop and D flip flop
Week 13	Learn about the working shift Register
Week 14	Learn about the working counters
Week 15	Learn how to draw complex logical circuits consisting of a group of logical circuits that have been studied in previous lectures
Week 16	**Preparatory week before the final exam.**

Delivery Plan (Weekly Lab. Syllabus): There is no Lab activities المنهاج الاسبوعي للمختبر:	
	Material Covered
Week 1	Lab1_ introduction to Multimedia Logic and crocodile program
Week 2	Lab2_ How to draw logical gates and form logical circuits
Week 3	Lab3_Simplify logical expressions and draw expressions before and after simplification, noting drawing outputs before and after simplification
Week 4	Lab4_Draw circles, half subtract full Adder, Half Adder, full subtract and observe the results of addition and subtraction
Week 5	Lab5-Draw multiplexer, demultiplexer, decoder, encoder logic circuits and how to connect them together in one logical circuit
Week 6	Lab6-Draw Filp_Flops logic circuits
Week 7	Lab7-Dealing with shift register and counter

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	1. principle of logic design (2020)by (Qasim mohammed Hussein)	Yes
Recommended Texts	Digital logic and computer Design by Morris Mano	No
Websites	Digital Logic And Computer Design By M. Morris Mano (2nd Edition).pdf - Google Drive	

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 programming			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☒ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	6				
SWL (hr/sem)	148				
Module Level	1		**Semester of Delivery**	2nd	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Harith A. Husain		**e-mail**		
Module Leader's Acad. Title	Assistant Lecturer		**Module Leader's Qualification**	master	
Module Tutor			**e-mail**		
Peer Reviewer Name	Harith Abdullah		**e-mail**		
Scientific Committee Approval Date	1/02/2026		**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture. 2- Develop skills in analyzing the usability of a web site. 3- Understand how to plan and conduct user research related to web usability. 4- Learn the core web technologies and programming languages that power the modern web. Starting with HTML and CSS and Javascript. 5- Learn some concepts in server-side programming using (PHP), along with MySQL database. 6- - Exploring how to create dynamic web applications that can process user data, store information, using content management systems (CMS).
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Understand the fundamental concepts of web development: Gain a solid understanding of how the web works, including client-server architecture, HTTP protocols, and the role of web browsers. 2. Develop proficiency in HTML and CSS: Acquire the skills to create well-structured web pages using HTML markup, apply CSS styles for layout and design, and create responsive and visually appealing websites. 3. Understand JavaScript programming: Learn the essentials of JavaScript programming, including variables, data types, and operators. 4. Build dynamic web applications: Explore server-side programming using PHP. Understand how to process user data, connect to databases, and generate dynamic content.

	5. Work with databases: Gain familiarity with database management systems like MySQL. Learn how to design and create database schemas. 6. Implement security measures: Understand common web security vulnerabilities and learn techniques to protect web applications from attacks. 7. Problem-solving and debugging skills: Develop the ability to identify and fix common issues in web development through effective troubleshooting and debugging techniques.
Indicative Contents المحتويات الإرشادية	1- Introduction to Web Development • Overview of web technologies and standards • Client-server architecture and HTTP protocols • Understanding web browsers and their rendering engines • HTML Fundamentals 2- Structure of HTML • HTML tags and elements • Working with text, images, links, and lists • Creating forms for user input • Semantic HTML and accessibility best practices 3- CSS Styling • CSS syntax and selectors • Box model and layout techniques • Applying styles to text, colors, backgrounds, and borders • CSS positioning and responsive design • CSS frameworks and libraries 4- JavaScript Basics • Introduction to JavaScript and its role in web development • Variables, data types, and operators 5- Server-Side Programming • Introduction to server-side programming languages (PHP)

	• Handling user input and form data • Working with databases (MySQL) • Templates for dynamic content • Session management and user authentication 6-Web Security • Common web security vulnerabilities • Guidelines and directions to protect the website

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Hands-on Practice: This Encourage students to actively engage in hands-on coding exercises and projects. Providing them with opportunities to apply theoretical concepts in practical scenarios, allowing them to gain proficiency through practice. 2. Project-Based Learning: Assigning projects that simulate real-world web development scenarios. This approach allows students to apply their knowledge and skills to create fully functional web applications, reinforcing their understanding and problem-solving abilities. 3. Code Review and Feedback: Incorporating code review sessions where students can share their code and receive constructive feedback. This process helps students identify areas for improvement, learn best practices, and enhance their coding style and techniques. 4. Collaborative Learning: Foster a collaborative learning environment where students can work together on group projects or problem-solving tasks. Encourage peer-to-peer discussions, code sharing, and knowledge exchange, as this can enhance understanding and expose students to diverse perspectives and solutions. 5. Online Resources and Documentation: Introduce students to reputable online resources, documentation, and tutorials related to web development. Teaching students how to effectively search for solutions, read and understand documentation, and leverage online communities and forums for support and learning.

	6. Practical Examples and Case Studies: Provide practical examples and case studies that demonstrate the application of web programming concepts in real-world scenarios. This helps students relate theoretical concepts to practical use cases, enhancing their understanding and problem-solving abilities. 7. Regular Assessments and Feedback: Conducting regular assessments, quizzes, and coding challenges to evaluate students' progress and understanding. Providing timely feedback to help students identify their strengths and areas that require improvement. 8. Continuous Learning and Exploration: Encouragement students to stay updated with the latest trends, tools, and technologies in web development. 9. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	2	10% (10)	3, 12	
	Projects	1	10% (10)	8-14	
	Report				
Summative	**Midterm Exam**	2 hr	20% (20)	11	

assessment	**Final Exam**	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction web programming
Week 2	Internet and Intranet, Web application ,web page, website , Classifying websites
Week 3	Client side script and server side scrip, Introduction to HTML
Week 4	HTML tags and attributes
Week 5	HTML - Titles and headings
Week 6	HTML – Lists
Week 7	HTML Images
Week 8	HTML – Tables
Week 9	HTML Frame and Form
Week 10	Introduction to CSS
Week 11	Internal and external CSS
Week 12	Introduction to JavaScript
Week 13	JavaScript Variables, data types, and operators
Week 14	Server-Side Programming (PHP with MySQL)
Week 15	Content Management Systems (CMS), Security vulnerabilities

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	HTML tags and attributes
Week 2	HTML – Titles, headings and Lists

Week 3	HTML Images
Week 4	**HTML – Tables**
Week 5	HTML Frame and Form
Week 6	Internal CSS
Week 7	External CSS
Week 8	Creating web page structure using HTML tags
Week 9	JavaScript variables definitions, data types
Week 10	JavaScript operators
Week 11	Implementing interactivity and dynamic content on web pages
Week 12	Install WAMP server
Week 13	Work with MySQL database
Week 14	Content Management Systems (CMS): install Joomla
Week 15	Create dynamic website

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	None	
Recommended Texts	Jon, Duckett. "HTML and CSS: Design and Build Websites." (2016).	
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	Artificial Intelligence			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code					
ECTS Credits	6				
SWL (hr/sem)	148				
Module Level	2		**Semester of Delivery**	2nd	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Qabas Amer Hameed		**e-mail**	Qabas.a.hameed@tu.edu.iq	
Module Leader's Acad. Title	lecturer		**Module Leader's Qualification**	Master	
Module Tutor			**e-mail**		
Peer Reviewer Name	Dr. Mohamed Aktham		**e-mail**		
Scientific Committee Approval Date	1/02/2026		**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	• Understanding the concept of AI and its applications • Understanding the problem state space and problem-solving. • Implementing and employing intelligent search methods to • solve problems that are not solved with traditional methods. • Using Heuristics in games. • Understanding the difference between blind search and heuristics search • To enable the learner to understand the basics of machine learning.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	• Understand the definition and application of AI • Understand the basic syntax and semantics of Python programming language. • Analyze and solve problems using Python programming. • Develop the student's thinking skills and the ability to analyze problems using logical programming principles. • Develop skills in debugging and testing AI search strategies in Python programming language. • • Providing the learner with the necessary skills to be able to build a learning model.
Indicative Contents المحتويات الإرشادية	• Fundamentals of AI • Problem in AI (Problem state space, Search space, and Problem-solving) • AI search taxonomy • Blind search algorithms • Heuristic search algorithms • the basics of machine learning

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain the definition of Artificial intelligence (AI), its applications, problem-solving techniques, types of search algorithms in AI, and the basic principles related to machine learning. This will provide students with a solid theoretical

	foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in learning. Discussions can include AI search algorithm examples, discussing practically how to program these algorithms, and exploring real-world applications of AI concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding AI projects. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming AI tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of how to program search strategies of AI and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using Python. 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and

	adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	9-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	AI Fundamental definitions, some techniques used today in AI, Some Applications of A
Week 2	Problem spaces; problem-solving by search
Week 3	Brute-force search (depth-first, breadth-first)
Week 4	Iterative deep DFS, Uniform cost first search

Week 5	Heuristic search (hill-climbing, best first search)
Week 6	A* algorithm
Week 7	Dijkstra algorithm
Week 8	Two-player games (minimax search, alpha-beta pruning)
Week 9	Introduction to Machine Learning
Week 10	Main Steps of ML
Week 11	**Midterm Exam**
Week 12	Supervised learning, Unsupervised learning, Reinforcement learning
Week 13	Methods for evaluating machine learning algorithms
Week 14	Project Due (Students Presentations part1)
Week 15	Project Due (Students Presentations part2)

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction to the basics of Python programming Language
Week 2	Introduction to the basics of Python programming Language
Week 3	Introduction to the basics of Python programming Language
Week 4	Running Examples of depth-first search
Week 5	Running Examples of breadth-first search
Week 6	Running examples of Uniform cost first search
Week 7	Running Examples of Best First Search

Week 8	Running Examples of A* search algorithm
Week 9	Running Examples of the Dijkstra Algorithm
Week 10	Exploring Python Libraries for ML: Discuss libraries like Numpy
Week 11	Exploring Python Libraries for ML: pandas
Week 12	Exploring Python Libraries for ML: matplotlib
Week 13	Exploring Python Libraries for ML: SciPy
Week 14	Exploring Python Libraries for ML: scikit-learn
Week 15	Answering students' questions and reviewing previous topics.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	"Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig (2020, 4th Edition) 2. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving" by George F. Luger (2021, 7th Edition)	No
Recommended Texts	Dimitris Varkas and Ioannis Pl. Vlashavos, " Artificial Intelligence for Advanced Problem-Solving Technique", published in the USA by Information science reference (an imprint of "IGI" Global),2008.	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	Relational databases and structured query language			**Module Delivery**	
Module Type	Core			☐ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	6				
SWL (hr/sem)	148				
Module Level	1		**Semester of Delivery**	2nd	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Mustafa Lateef Fadhil		**e-mail**	Mustafa.l.fadhil@tu.edu.iq	
Module Leader's Acad. Title	Assistant Lecturer		**Module Leader's Qualification**	Master	
Module Tutor			**e-mail**		
Peer Reviewer Name	Mohamed Aktham		**e-mail**		
Scientific Committee Approval Date	1/02/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Relational databases and structured query language	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. Understanding Relational databases and structured query language: The module aims to provide students with a deep understanding of relational databases, structured query language, and their manipulation techniques. Students will learn about data modeling using the entity–relationship (ER) model. 2. Mastery of Entity: The module aims to develop students' proficiency in using entity diagram in Microsoft SQL Server. Students will learn the concepts of Entity Types, Entity Sets, Attributes, and Keys. 3. Data Handling and Structured SQL Query Operations: The module aims to introduce students to file handling concepts and techniques in relational database. Students will learn how to Relationship Types, Relationship Sets, Roles, and Structural Constraints. 4. Data Management and Organization: The module aims to teach students how to manage and organize data effectively in database. They will learn to create, update, and delete data, organize data directories, and handle data-related operations. Students will understand the importance of data management in real-world database scenarios. 5. Practical Application and Problem-Solving: Throughout the module, students will be exposed to practical The Enhanced Entity–Relationship (EER) Model exercises and problem-solving tasks. They will apply the concepts learned to solve real-world programming challenges, consolidating their understanding and enhancing their problem-solving skills. By focusing on the relational algebra, relational calculus and data handling in Microsoft SQL server, this advanced module aims to provide students with a comprehensive understanding of these concepts and their practical application. Students will develop the skills necessary to manipulate complex data , handle data, and write efficient and reliable code.

Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Demonstrate an in-depth understanding of Relational Database Design by ER- and EER-to-Relational Mapping and data handling concepts in SQL. 2. Apply advanced operations, such as Update Operations, Transactions, and Dealing with Constraint Violations. 3. Manipulate Relational Calculus like Unary Relational Operations, Binary Relational Operations and Relational Algebra Operations from Set Theory. 4. Utilize pointers proficiently for data manipulation, including memory addresses, and pointer arithmetic 5. Manage and organize data effectively, including creating, updating, deleting, and organizing data directories. 6. Apply efficient programming techniques, optimize code, and adhere to best practices for writing clean and readable code. 7. Demonstrate problem-solving skills by applying the learned concepts to solve real-world programming challenges. 8. Work collaboratively in teams, communicate effectively, and share knowledge and ideas related to advance programming concepts. 9. Adapt to new programming concepts and technologies beyond the scope of the course, building a foundation for lifelong learning in programming. These learning outcomes reflect the knowledge, skills, and competencies that students will acquire upon completing the advanced programming course. The outcomes emphasize both theoretical understanding and practical application, preparing students for real-world programming challenges and further studies in the field of computer science.
Indicative Contents المحتويات الإرشادية	1. Review of Basic Database Concepts: - Recap of fundamental database concepts, including variables, data types, control structures, and functions. 2. Data Modeling Using the Entity–Relationship (ER) Model: - Using High-Level Conceptual Data Models for Database Design - A Sample Database Application - Entity Types, Entity Sets, Attributes, and Keys - Relationship Types, Relationship Sets, Roles, and Structural Constraints - Weak Entity Types 3. The Enhanced Entity–Relationship (EER) Model: - Subclasses, Super classes, and Inheritance - Specialization and Generalization 4. Relational Model Concepts: - Domains, Attributes, Tuples, and Relations

	- Characteristics of Relations

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain database concepts, SQL syntax, and techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing database best practices, and exploring real-world applications of database concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on database exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on database exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning.

	d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of database programming concepts and encourage independent problem-solving. These assignments may involve implementing sql query, designing database systems and tables. 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding database programming concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	5-14	
	Report	1			
Summative	**Midterm Exam**	2 hr	10% (10)	11	

assessment	**Final Exam**	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Recap of fundamental database concepts, including variables, data types, control structures, and functions.
Week 2	Data Modeling Using the Entity–Relationship (ER) Model
Week 3	Using High-Level Conceptual Data Models for Database Design
Week 4	A Sample Database Application
Week 5	Relationship Types, Relationship Sets, Roles, and Structural Constraints
Week 6	Weak Entity Types
Week 7	The Enhanced Entity–Relationship (EER) Model
Week 8	ER Diagrams, Naming Conventions, and Design Issues
Week 9	Example of Other Notation: UML Class Diagrams
Week 10	The Enhanced Entity–Relationship (EER) Model
Week 11	Constraints and Characteristics of Specialization and Generalization Hierarchies
Week 12	Modeling of UNION Types Using Categories
Week 13	The Relational Data Model and Relational Database Constraints
Week 14	Second Project Due (Students Presentations part1)
Week 15	Second Project Due (Students Presentations part1)

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Modeling of UNION Types Using Categories

Week 2	Running Examples on Unary Relational Operations: SELECT and PROJECT
Week 3	Relational Algebra Operations from Set Theory: **The UNION, INTERSECTION, and MINUS Operations**
Week 4	**Binary Relational Operations:** **JOIN and DIVISION**
Week 5	Running Examples on unary and binary
Week 6	Running Examples on set theory
Week 7	Additional Relational Operations
Week 8	Examples of Queries in Relational Algebra
Week 9	The Tuple Relational Calculus
Week 10	Running More Examples on Tuple Variables and Range Relations
Week 11	Running Examples on Database and tables
Week 12	Running More Examples on database
Week 13	Running More Advanced Programs on data
Week 14	Wrapping up
Week 15	Answering Students Questions and Extra Advising on Real World Application Programming

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Elmasri, R. (2021). Fundamentals of database systems seventh edition.	Yes
Recommended Texts		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	English Language(3)			**Module Delivery**	
Module Type	s			☒ **Theory** ☐ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	2				
SWL (hr/sem)	50				
Module Level		3	**Semester of Delivery**		5
Administering Department			**College**		
Module Leader	Ahmed Faeq		**e-mail**		
Module Leader's Acad. Title		Asst. lecturer	**Module Leader's Qualification**		master
Module Tutor	None		**e-mail**	None	
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date		1/2/2026	**Version Number**		1.0

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. To be able to speak English fluently and accurately. 2. To think in English and then speak. 3. To be able to talk in English. 4. To be able to compose freely and independently in speech and writing. 5. To be able to read books with understanding.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Important: Write at least 6 Learning Outcomes, better to be equal to the number of study weeks. 1. To address grammar issues that students encounter in their daily speech, writing, reading and listening 2. To address the issue of grammatical errors that affect effective communication 3. To improve your reading skills through the practice of vocabulary enrichment, reading comprehension exercises, speed reading strategies, written responses, discussions, and reflections 4. Recognize the structure and organization of paragraphs, 5. Use strategies to think critically about reading and use appropriate technology to enhance reading comprehension, reading speed, and vocabulary development 6. Develop the writing skill.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. Introduction, Study material review [1 hr] Grammar: Have (got) to, practices. Should/must, questions and answers. Reading. Vocabulary: words that go together, everyday English at the doctors. [6hrs] Time and conditional clauses, practices (when, as soon as). listening and speaking/ life in 2050, Reading and speaking/ the world's first megalopolis. Vocabulary: Hot verbs/ take- get- do and make. [7 hrs] Grammar: verb patterns and infinitives, practices. Vocabulary: -ed/ -ing adjective, reading about (Into the wild). Expressions about exclamations with so and such. [8 hrs] Grammar: actives and passives voice, practices. Verbs and nouns that go together, practices. Reading: about the discovery of DNA., expressions

	about(notices). [6 hrs] Review the study units. [2 hrs]

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	- The main strategy that will be adopted in developing the four skills: The skill of speaking, The skill of reading, The skill of writing, The skill of listening, Also, enable the students for the use of grammar correctly,

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	32	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	18	**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**	3	15% (15)	3,6 and 10	LO #1, #2 and #4

assessment	**Assignments**	3	15% (15)	2,8 and 12	LO #3, #5 and #6
	Report	1	10% (10)	13	LO #2, #4 and #6
Summative assessment	**Midterm Exam**	1hr	10% (10)	7	LO #1 - #5
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	Introduction: about study materials of Headway Pre-intermediate Plus.
Week 2	Grammar: Have (got) to, practices.
Week 3	Should/must, questions and answers. Reading.
Week 4	Vocabulary: words that go together, everyday English at the doctors.
Week 5	Grammar: verb patterns and infinitives, practices.
Week 6	Time and conditional clauses, practices (when, as soon as). listening and speaking/ life in 2050.
Week 7	Mid-term Exam
Week 8	Reading and speaking/ the world's first megalopolis.
Week 9	Vocabulary: Hot verbs/ take- get- do and make.
Week 10	Vocabulary: -ed/ -ing adjective, reading about (Into the wild).
Week 11	Expressions about exclamations with so and such.
Week 12	Grammar: actives and passives voice, practices.
Week 13	Verbs and nouns that go together, practices.
Week 14	Reading: about the discovery of DNA., expressions about(notices).
Week 15	Study material review
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	None
Week 2	None

Week 3	None
Week 4	None
Week 5	None
Week 6	None
Week 7	None

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Headway pre-intermediate plus student's book(john and Lize Soars)	Yes
Recommended Texts	Headway pre-intermediate plus work's book	Yes
Websites	https://7esl.com/ , https://worldenglishblog.com/hasnt-vs-doesnt-have-pdf/	

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	Visual Programming			**Module Delivery**	
Module Type				☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code					
ECTS Credits					
SWL (hr/sem)	148				
Module Level		2	**Semester of Delivery**		2nd
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Israa Rafaa Abdulkader		**e-mail**	Israa.R.Abdalkader@tu.edu.iq	
Module Leader's Acad. Title		lecture	**Module Leader's Qualification**		Master
Module Tutor			**e-mail**		
Peer Reviewer Name		Mohamed Aktham	**e-mail**		
Scientific Committee Approval Date		1/02/2026	**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	5. Abstraction: hiding the implementation details and showing only functionality to the user. 6. Encapsulation: data hiding. 7. Polymorphism: the ability to take more than one form. 8. Inheritance: objects of one class acquire the properties of objects of another class.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. Codes basic programs in java programming language 2. Introduction (Object Oriented Programming Characteristics, OOP Definition, OOP Concepts, Differences from Procedure Oriented Programming (POP)). 3. Classes and Objects (methods, properties). 4. Constructors and Destructors. 5. Inheritance (Single Inheritance, Multilevel Inheritance) and Polymorphism (Function Overloading, Operator Overloading, Virtual Function). 6. Abstract Classes and Methods, Sealed Classes and Interfaces, Generic types and methods and collection class.
Indicative Contents المحتويات الإرشادية	Basics of java Characteristics of Object-Oriented Programming Classes and Objects Working with Constructors Data Members Using Static Variables & Understanding Scope Overloading Inheritance Exceptions and Errors

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain programming concepts, java syntax, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows

	them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively. 4. Programming Assignments: Assignments will be given to students to reinforce their understanding of programming concepts and encourage independent problem-solving. These assignments may involve implementing algorithms, designing software systems, or developing small-scale projects using java . 5. Code Reviews and Feedback: The instructor will provide feedback on students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for

	individual consultations and provide support to students who need additional help or guidance in understanding programming concepts or completing assignments.

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

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation
تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	9-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Recap of fundamental programming concepts, including variables, data types, control structures Using Java language
Week 2	Introduction to method(function)
Week 3	Mathematical Functions and Class Character
Week 4	Introduction to String and methods

Week 5	Introduction to Arrays (Linear arrays) and Searching and Sorting Linear Arrays
Week 6	Multidimensional Arrays and Square Arrays
Week 7	Introduction (Object Oriented Programming Characteristics, OOP Definition, OOP Concepts, Differences from Procedure Oriented Programming (POP)).
Week 8	Classes and Objects (methods, properties).
Week 9	Polymorphism (Overloading and Overriding Methods) & Constructors.
Week 10	Inheritance (Single Inheritance, Multilevel Inheritance).
Week 11	**Midterm Exam**
Week 12	Abstract Classes and Methods, Interfaces, Generic types and methods..
Week 13	Array List Class
Week 14	Project Due (Students Presentations part1)
Week 15	Project Due (Students Presentations part2)

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Getting used to java netbeans Interfaces and practicing some commands
Week 2	Running Examples on method
Week 3	Running examples on Mathematical Functions and Class Character
Week 4	Running Examples on Strings
Week 5	Practicing Arrays further (Searching and Sorting)
Week 6	Running Examples on 2D and Square Arrays
Week 7	Running Examples on Object Oriented Programming Characteristics

Week 8	Running Examples Classes and Objects (methods, properties).
Week 9	Running Examples on Overloading , Overriding Methods and Constructors.
Week 10	Running More Examples on Inheritance (Single Inheritance, Multilevel Inheritance).
Week 11	Running Examples on classes
Week 12	Running More Examples on Abstract Classes and Methods, Interfaces, Generic types and methods.
Week 13	Running examples on Array List Class
Week 14	Running programs of advanced of class
Week 15	Answering students' questions and reviewing previous topics.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Herbert Scildt, Java The complete reference Eleventh edition	No
Recommended Texts		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	Probability and Statistics			**Module Delivery**	
Module Type				☒ Theory ☒ Lecture ☒ Lab ☐ Tutorial ☐ Practical ☐ Seminar	
Module Code					
ECTS Credits	6				
SWL (hr/sem)	150				
Module Level		**UG I**	**Semester of Delivery**		**2**
Administering Department		**Computer Science**	**College**	**CSM**	
Module Leader	Kahlan Faiq Hasan		**e-mail**	kahlan.aljuburi@tu.edu.iq	
Module Leader's Acad. Title		**Teacher Assistant**	**Module Leader's Qualification**		**Ms**
Module Tutor	**Rafal Salih**		**e-mail**	**Rafalsaleh@tu.edu.iq**	
Peer Reviewer Name			**e-mail**		
Scientific Committee Approval Date		**1/2/2026**	**Version Number**		**1.0**

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	1. Mathematics for Computer Science 2. Calculus 3. Linear Algebra 4. Discrete Mathematics 5. Introduction to Programming	**Semester**	
Co-requisites module	1. **Artificial Intelligence and Machine Learning**	**Semester**	

	2. **Data Science and Big Data** 3. Algorithms and Optimization 4. Computer Vision and Image Processing 5. Cryptography and Network Security 6. Computer Networks and Distributed Systems 7. Wireless Communication and Signal Processing 8. Queuing Theory in Network Design 9. Network Simulations 10. Internet of Things (IoT) and Sensor Networks		

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Objectives أهداف المادة الدراسية	1. Clarification of the fundamental concepts of probability. 2. Understanding Bayes' theorem and conditional probability. 3. Learning about discrete distributions and how to find expectation and variance. 4. Understanding continuous distributions and how to find expectation and variance, as well as studying the associated theorems. 5. Studying probability mass function (PMF), probability density function (PDF), cumulative distribution function (CDF).
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. The student should state the basic definitions. 2. Introduction and basic definitions of the topic of probability. 3. Distinguish between the probability mass function (PMF), probability density function (PDF), cumulative distribution function (CDF) with various examples. 4. Understand the joint random variables and dependent and independent random variables.

	5. Recognize discrete distributions, their types, and study examples with theorems on expectation and variance. 6. Student can compute conditional probabilities based on the definition of conditional probability 7. Understand the continuous distributions and their types. 8. To be able to distinguish between discrete distributions and continuous distributions. 9. To be able to distinguish between discrete random variables and continuous random variables.
Indicative Contents المحتويات الإرشادية	1. Introduction and basic definitions of the topic of probability. 2. Study of Bayes' theorem and conditional probability. 3. Understanding discrete distributions and their types. 4. Studying examples of discrete distributions with theorems on expectation and variance. 5. Understanding continuous distributions, with focus on the normal and standard normal distributions, accompanied by examples.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The primary strategy to be adopted in delivering this unit is to encourage students to engage in exercises while simultaneously enhancing and expanding their critical thinking skills. This will be achieved through interactive classes and tutorials, as well as by considering types of simple experiments that involve some sampling activities of interest to students.

Student Workload (SWL) الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	78	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	5.2

Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	72	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	4.8
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 التقييم التكويني	**Quizzes**	10	20% (20)	3-14	
	Assignments	2	10% (10)	6 , 12	
	Projects / Lab.				
	Report				
Summative assessment التقييم التلخيصي	**First Midterm Exam**	2hr	10% (10)	6	
	Second Midterm	2hr	10% (10)	12	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

	Material Covered
Week 1	**Module 1** - Introduction to statistics and probability - Importance of statistics in computer science. - Fundamental concepts: population, sample, parameters. - Mean, Median, Mode. - Application examples in computer science.

Week 2	**Module 1** - Probability space - Kolmogorov's axioms - Conditional probability
Week 3	**Module 1** - Independence - The law of total probability - Bayes' rule
Week 4	**Module 1** - Random variables (discrete, continuous, mixed) - Probability Mass Function (pmf)
Week 5	**Module 1** - Cumulative Distribution Function (cdf) - Probability Distribution Function (pdf)
Week 6	***First MIDTERM EXAMINATION***
Week 7	**Module 1** - Expected value, moments - Variance, and standard deviation - Conditional Distributions and Moments
Week 8	**Module 1** Important distributions: Definitions, Parameters, and Properties. - Discrete Uniform - Bernoulli - Binomial
Week 9	**Module 1** Important distributions: Definitions, Parameters, and Properties. - Geometric - Poisson - Continuous Uniform
Week 10	**Module 1** Important distributions: Definitions, Parameters, and Properties. - Exponential - Gaussian
Week 11	***Second MIDTERM EXAMINATION***
Week 12	**Module 2** Bounding Properties - Markov inequality - Chebyshev inequality - Chernoff bounds
Week 13	**Module 2** Random Vectors - Joint pmf - Marginal pmf

	- Conditional Distributions - Independence of random variables
Week 14	**Module 2** - Joint moments of random vectors - Correlation, Covariance
Week 15	**Module 2** - Correlation Coefficient - Correlation and Covariance matrices
Week 16	***Final Examination***

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر			
	Material Covered		
Week 1	**Introduction to MATLAB for Probability and Statistics** • Setting up the environment • Basic operations and commands	**Week 8**	**EXAM**
Week 2	**Introduction to MATLAB for Probability and Statistics** • Setting up the environment • Basic operations and commands	**Week 9**	**Random Variables and Simulations** • Defining and working with discrete and continuous random variables • Simulating random experiments
Week 3	**Descriptive Statistics with MATLAB** • Calculating mean, median, and mode • Visualizing data distributions and their functions	**Week 10**	**Random Variables and Simulations** • Defining and working with discrete and continuous random variables • Simulating random experiments
Week 4	**Descriptive Statistics with MATLAB** • Calculating mean, median, and mode • Visualizing data distributions and their functions	**Week 11**	**Random Variables and Simulations** • Defining and working with discrete and continuous random variables • Simulating random experiments
Week 5	**Probability Distributions in MATLAB** • Generating and plotting common probability distributions (e.g., POISSON, EXPONENTIAL, NORMAL) • Calculating probabilities pdf and cdf	**Week 12**	**Random Variables and Simulations** • Defining and working with discrete and continuous random variables • Simulating random experiments
Week 6	**Probability Distributions in MATLAB** • Generating and plotting common probability distributions (e.g., POISSON, EXPONENTIAL, NORMAL) • Calculating probabilities pdf and cdf	**Week 13**	A comprehensive review

Week 7	**Probability Distributions in MATLAB** • Generating and plotting common probability distributions (e.g., POISSON, EXPONENTIAL, NORMAL) • Calculating probabilities pdf and cdf	**Week 14**	EXAM
		Week 15	A comprehensive review

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts النصوص المطلوبة	- Probability, Statistics, and Random Processes for Electrical Engineering Alberto Leon-Garcia Pearson, 3rd Edition	NO
Recommended Texts	- Introduction to probability models, Sheldon M. Ross, 9th Edition - Intuitive Probability and Random Processes using MATLAB, Steven Kay	No
Websites	https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2022/	

Grading Scheme
مخطط الدرجات

Group	Grade	التقدير	Marks %	Definition

Success Group (50 - 100)	**A -** Excellent	امتياز	90 - 100	Outstanding Performance أداء مذهل
	B - Very Good	جيد جدا	80 - 89	Above average with some errors فوق المتوسط مع بعض الأخطاء
	C - Good	جيد	70 - 79	Sound work with notable errors العمل السليم مع أخطاء ملحوظة
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings عادل ولكن مع نواقص كبيرة
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria العمل يلبي الحد الأدنى من المعايير
Fail Group (0 – 49)	**FX –** Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded مطلوب المزيد من العمل ولكن الائتمان الممنوح
	F – Fail	راسب	(0-44)	Considerable amount of work required قدر كبير من العمل المطلوب

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	Machine Learning			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code					
ECTS Credits	4				
SWL (hr/sem)	148				
Module Level	2		**Semester of Delivery**	3rd	
Administering Department	Computer Science		**College**	CCSM	
Module Leader	Armaneesa Naaman Hasoon		**e-mail**	armaneesa@tu.edu.iq	
Module Leader's Acad. Title	Assistant Professor		**Module Leader's Qualification**	Master	
Module Tutor			**e-mail**		
Peer Reviewer Name	Dr. Mohammed Aktham		**e-mail**		
Scientific Committee Approval Date	1/02/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Artificial intelligence	**Semester**	
Co-requisites module	Python Calculus, Probability	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	•Understand the term machine learning and its applications •Understand the steps of machine learning. •Study the types of learning •Solve problems that traditional methods cannot solve.

	•Understand the methods of obtaining and processing data •Understand the methods of extracting features • Train and evaluate machine learning models
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	•Understand the definition and applications of machine learning •Understand data processing methods. •Develop the student's thinking skills and the ability to build a machine learning model used for prediction or classification •Provide the learner with the skills necessary to be able to evaluate the performance of the machine learning model and apply it in real-time
Indicative Contents المحتويات الإرشادية	• Machine Learning Basics

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1 .Lectures: The instructor will give lectures to introduce and explain the definition of machine learning methods and their applications. 2 .Interactive discussions: Engaging students in interactive discussions allows them to ask questions, seek clarification, and actively participate in learning. Discussions can include examples of machine learning algorithms, discuss how these algorithms are programmed in practice, and explore real-world applications of AI concepts. 3 .Lab: Lab sessions are dedicated, hands-on sessions where students apply the concepts they have learned in lectures to practical programming exercises. Key strategies for lab sessions include: a. Programming exercises: Students will work on programming exercises and projects in the lab, providing them with hands-on experience in programming machine learning projects. b. Guided training: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and peer learning: Students can collaborate with their peers in the lab, promoting teamwork and enabling knowledge sharing. Working together on machine learning project programming assignments promotes discussion, problem solving, and peer learning.

	D. Access to Equipment and Resources: The lab should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the resources needed to complete lab exercises and assignments effectively. 4 .Programming Assignments: Students will be given assignments to enhance their understanding of how to program machine learning models to encourage independent problem solving. These assignments may include implementing algorithms, designing software systems, or developing small-scale projects using Python. 5 .Code Reviews and Feedback: The instructor will provide feedback on students' code, review their solutions, and make suggestions for improvement. This feedback will help students improve their coding skills and adhere to best practices. 6 .Office Hours and Individual Support: The instructor should provide individual consultations and support for students who need additional help or guidance in understanding and completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	9-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Introduction to machine learning
Week 2	Steps of a Machine Learning Model in Detail
Week 3	Performance evaluation metrics for machine learning methods
Week 4	Regression models
Week 5	Simple linear regression
Week 6	Multi-linear regression.
Week 7	Polynomial linear regression
Week 8	Mid Exam
Week 9	Classification Models: Logistic Regression
Week 10	Decision tree
Week 11	Random forest
Week 12	Support vector machine
Week 13	Clustering: Introduction to Clustering, K-Means, Hierarchical Clustering
Week 14	Project Due (Students Presentations part1)
Week 15	Project Due (Students Presentations part2)

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	**Material Covered**
Week 1	Introduction to the basics of preprocessing a dataset
Week 2	Introduction to the basics of Evaluation ml models
Week 3	Build a Simple Linear Regression using Scikit-learn.
Week 4	Build a multivariable linear Regression using Scikit-learn.
Week 5	Build a polynomial regression using Scikit-learn.

Week 6	Build a logistic regression classifier using Scikit-learn.
Week 7	Build a Decision tree classifier using Scikit-learn.
Week 8	Build a Random Forest classifier using Scikit-learn.
Week 9	Build a Support vector machine classifier using Scikit-learn.
Week 10	Build a K-Means clustering method using Scikit-learn.
Week 11	Build a hierarchical clustering using Scikit-learn.
Week 12	Build a hierarchical clustering using Scikit-learn.
Week 13	Quiz
Week 14	Answering students' questions and reviewing previous topics.
Week 15	Answering students' questions and reviewing previous topics.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition by Aurélien Géron (Author), 2017.	No
Recommended Texts		
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					
ECTS Credits					
SWL (hr/sem)	125				
Module Level		3	**Semester of Delivery**		1
Administering Department		Computer Science	**College**	Computer Science and Mathematics	
Module Leader	**Lecturer. Mohanad Dawood Salman**		**e-mail**	mohanaddawoodalroomi@tu.edu.iq	
Module Leader's Acad. Title		**Lecturer.**	**Module Leader's Qualification**		M.S.C
Module Tutor	**Assistant Lec.: Luay Ibrahim Klalif**		**e-mail**	luay.i.khalaf@tu.edu.iq	
Peer Reviewer Name		Mohamed Aktham	**e-mail**		
Scientific Committee Approval Date		1/2/2026	**Version Number**		1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Basic and advanced programming language Computational theory Computer architecture	**Semester**	1 2 2
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	9. Student be able to Identify principles, techniques and tools for compilers of programming languages. 10. Student be able to design a compiler for a (simplified) (programming) language. 11. Student know how to use compiler construction tools, such as generators of scanners and parsing. 12. Student be familiar with assembly code and virtual machines. 13. Be able to define LL(1), LR(1), and SLR(1) grammars 14. Student be familiar with compiler analysis and optimization techniques.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	**After completing this module, a student should be able to:** 1. Understand the principles guiding all aspects of the compilation process. 2. Recognize the function of each of the fundamental components of a conventional compiler. 3. Demonstrate familiarity with compiler front-end concepts, such as lexical analysis, syntactic analysis, semantic analysis, type checking, symbol tables, and so on. 4. Identify the type of compiler errors that appear during programming and analyze the errors. 5. Review and practical application of the C++ programming language. 6. Finally, the student will build the front-end parts of his or her own compiler using a higher language.
Indicative Contents المحتويات الإرشادية	Indicative content includes the following. <u>Part A - Front End</u> - Compiler phases and overall working, Scanners (Regular Expressions, NFA/DFA, Scanner Generators). [10 hrs] - Parsers (Grammars, Ambiguity Removal, LL, LR, and other deterministic parsing).[10 hrs] - Semantic Analysis (Symbol Tables, Syntax Driven Analysis, Type Systems, Attribute Grammars). [10 hrs] Revision problem classes [5 hrs] <u>Part C –Middle End</u> - Intermediate Forms, Syntax Directed Translation, Translation of: Expressions, Control Structures and Back-patching, Function and Procedure Calls, Runtime Organization. [10 hrs] Revision problem classes [5 hrs] <u>Part B - Back-end</u>

	- Code optimization: code is made shorter and simpler by combining steps or removing unnecessary steps which lead to the generation of optimized code. . [10 hrs] - Code generation: coverts the intermediate code or the optimized code into the target language. Usually, the target language is the machine code. [5 hrs] - Selected, allocated all available memory and registers, then run to process inputs and provide results. [5 hrs]

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	• Using active learning: This type of learning includes engaging learners in active and interactive learning processes, such as discussions, practical experiments, educational games, and knowing the students' scientific level by evaluating their test results. • Software project: Emphasize design patterns, teamwork, and programming methodology by constructing a compiler to meet assigned specifications. • Technology-based learning: includes the use of technology in learning and teaching processes, such as the use of multimedia, educational applications, and online educational platforms. Technology can help improve access to knowledge and enhance interaction and engagement. • Application of theory: Emphasize the role of theory to enable automation of compiler tasks, and illustrate the limitations of that theory.

Student Workload (SWL) الحمل الدراسي للطالب			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	75	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	5
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	50	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	3.3
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	3	10% (10)	5, 9,12	LO #1-4
	Assignments	5	10% (10)	4,5,8,9,12	LO # 2- 5
	Projects / Lab.	1	10% (10)	Continuous	
	Report	0			
Summative assessment	Midterm Exam	2 hr	10% (10)	7	LO # 1-7
	Final Exam	3 hr	60% (60)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered
Week 1	▪ Programing languages. Introduction to compiler.
Week 2	▪ Language-processing system (Compilation). The Structure of Compiler.
Week 3	▪ The phases of compiler design.
Week 4	▪ Lexical analysis phase. Type of errors for Lexical analysis phase
Week 5	▪ Syntax definition. Context free grammar.
Week 6	▪ Parsing tree & Leftmost and Rightmost derivations.
Week 7	▪ Mid-term Exam
Week 8	▪ syntax analysis phase.
Week 9	▪ Problems of compiler for syntax analysis phase. ▪ regular expression and Regular Definitions
Week 10	▪ Top –down parsing .
Week 11	▪ bottom up parsing.
Week 12	▪ Ambiguous Grammars
Week 13	▪ Parser generators.
Week 14	▪ Exam Parser generators.
Week 15	▪ Semantic analysis phase. ▪ Type of errors for semantic analysis.
Week 16	▪ Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

المنهاج الاسبوعي للمختبر

	Material Covered
Week 1 , 2	Lab 1: Array characters in C++ and related functions. String characters in C++ and related functions.
Week 3 , 4	Lab 2: program Lexical analysis phase of second tasks.
Week 5 , 6	Lab 3: program to delete comment in one line. program to delete to comment in more lines. program of compaction of consecutive white spaces into one.
Week 6	Lab 4: program to build the Lexemes table.
Week 7	Mid-term Exam
Week 8 , 9	Lab 5: Programs of transition diagram
Week 10 , 11 , 12 , 13	Lab 6: Program checks keywords. Program checks identifiers. Program check the number. Program checks mathematical operations. Program check the number. Program checks mathematical operations.
Week 14 , 15	Type Checking
Week 16	Preparatory week before the final Exam.

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Basics of Compiler Design, Torben E. Mogensen, 2009.	Yes
Recommended Texts	Compilers , principles , Techniques and tools by Aho,Lam, Sethi and Ullman, 2nd Ed. Addison – Wesely , 2006. من البداية إلى البرمجة الكيانية ، الدكتور المهندس .نضال خضير العبادي C++ .(2011)	Yes
Websites	• http://lectures.iugaza.edu.ps/playlist/145/%D8%AA%D8%B5%D9%85%D9%8A%D9%85-%D8%A7%D9%84%D9%85%D8%AA%D8%B1%D8%AC%D9%85%D8%A7%D8%AA-compiler-design	

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	Operating System 2			**Module Delivery**	
Module Type	Core			☐ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	8				
SWL (hr/sem)	200				
Module Level		1	**Semester of Delivery**		1st
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Ayad Ahmed Yass		**e-mail**	ayad.a.yass@tu.edu.iq	
Module Leader's Acad. Title		Lecturer	**Module Leader's Qualification**		Phd
Module Tutor	Huthifa Mohammed		**e-mail**	huthife@tu.edu.iq	
Peer Reviewer Name		Mahammed Aktham	**e-mail**		
Scientific Committee Approval Date		1/2/2026	**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	Understanding basics of Operating system	**Semester**	2

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To describe the advance of computer systems 2. To describe the services an operating system provides to users, processes, and other systems. 3. To develop students' understanding skills for various operating system 4. To help students to deal with computer components. 5. To provide students with hands-on experience in Linux command line 6. To advance the students skills to working with scripting to deal with operating system. 7. To enhance students' ability to troubleshooting operating system problems. 8. To level up the students' skills so they be able to enter the market places since the OS consider the most required in most organizations
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Upon successful completion of this module, students should be able to: 1. Demonstrate a solid understanding of the fundamental principles and concepts of operating system. 2. to help the student to understand the components and security of the operating system. 3. At this point, the student will be able to deal with different operating system 4. Enhance the operating system components and avoid the security issues 5. Student able to work with Linux command as the Linux is the most required in the market nowadays 6. Less work and increase productivity by learning the scripting 7. fixing and troubleshooting all OS related issues 8. leveling up the skills means now the students can enter any market place without hesitating
Indicative Contents المحتويات الإرشادية	13- To describe the basic organization of computer systems. 14- To describe the services an operating system provides to users, processes, and other systems 15- To discuss the various ways of structuring an operating system 16- To introduce the notion of a process and a thread

	17- To introduce CPU scheduling, which is the basis for multi programmed operating systems 18- To develop a description of deadlocks

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain operating concepts, like processes, threads, memory management, file systems, and system calls. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing database best practices, and exploring real-world applications of operating system 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on database exercises. Key strategies for the laboratory sessions include: a. Linux Exercises: Students will work on linux exercises and projects in the laboratory, providing them with practical experience in command line and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' skills They can help students debug their scripting, identify errors, and improve their command line skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on Linux tasks promotes discussions, problem-solving, and peer learning. d. Equipment and Resource Access: The laboratory should provide access to computers, necessary software tools, programming references, and relevant online resources. This ensures that students have the necessary resources to complete their lab exercises and assignments effectively.

	4. command line Assignments: Assignments will be given to students to reinforce their understanding of command line concepts and encourage independent problem-solving. These assignments may involve implementing scripting. 5. Reviews and Feedback: The instructor will provide feedback on students' scripting, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices. 6. Office Hours and Individual Support: The instructor should be available for individual consultations and provide support to students who need additional help or guidance in understanding command concepts or completing assignments.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	93	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	107	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	7.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation

تقييم المادة الدراسية

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

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
Week No.	**Material Covered**
Week 1	Process Synchronization Part1
Week 2	Process Synchronization Part2
Week 3	Critical Section Problems part1
Week 4	Critical Section Problems part2
Week 5	Deadlock
Week 6	Method of Handling Deadlock
Week 7	Deadlock Avoidance part1
Week 8	Deadlock Avoidance part2
Week 9	Deadlock Detection part1
Week 10	Deadlock Detection part2
Week 11	Main Memory Part1
Week 12	Main Memory Part2
Week 13	Segmentation
Week 14	Segmentation and Memory Allocation
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Introduction about the advance Linux usage
Week 2	Advance Linux Commands
Week 3	Explain about the App that using Linux
Week 4	**Introduction about the bash scripting**

Week 5	Linux Shell Scripting
Week 6	How to use shell scripting
Week 7	Advance shell scripting command
Week 8	Linux command line
Week 9	Shell Scripting Excustion
Week 10	Installing software in Linux
Week 11	Understanding Linux logs
Week 12	Identifying issues from the logs
Week 13	Shell scripting comparisons with normal command line
Week 14	Troubleshooting issues
Week 15	Applying instructor's feedback on students' projects

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	*Abraham* Silberschatz (2022). Operating System Concept nine edition.	Yes
Recommended Texts		
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	Multimedia and Network Security			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits					
SWL (hr/sem)	148				
Module Level	4	**Semester of Delivery**	2nd		
Administering Department	Computer Science	**College**	CCSM		
Module Leader	Maath Frman Ismaeel	**e-mail**	Maath.f.ismaeel@tu.edu.iq		
Module Leader's Acad. Title	Assistant Lecturer	**Module Leader's Qualification**	MSc.		
Module Tutor		**e-mail**			
Peer Reviewer Name	Mohamed Aktham	**e-mail**			
Scientific Committee Approval Date	02/01/2026	**Version Number**	1.0		

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	11. **Foundational Network Defense:** To establish a deep understanding of network security principles, focusing on the protection of data integrity, confidentiality, and availability across different network infrastructures. 12. **Authentication & Identity Management:** To evaluate and implement robust methods for key distribution and user authentication to ensure only authorized entities access network resources. 13. **Infrastructure & Cloud Protection:** To analyze the security challenges unique to modern environments, including **Cloud Security**, **Wireless Networks**, and **Network Access Control (NAC)**. 14. **Protocol Analysis:** To master the technical specifics of transport-level security (TLS/SSL) and IP-level security (IPsec) to secure communication channels. 15. **Multimedia Data Security:** To introduce the concepts of multimedia security, moving beyond standard text-based encryption to handle the complexities of image, audio, and video data. 16. **Information Hiding Techniques:** To differentiate between and apply the principles of **Steganography** (hiding the existence of a message) and **Digital Watermarking** (protecting ownership and integrity). 17. **Practical Security Programming:** To develop hands-on proficiency using **Python** for security tasks, including network reconnaissance, key exchange, and the algorithmic implementation of watermarking. 18. **Media Authentication & Copyright:** To explore the technical methods used for embedding and extracting watermarks in various media types (Grey/Color images, Audio, and Video) for copyright protection.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. **Analyze Network Vulnerabilities:** Explain the fundamental threats to network security and the mechanisms used to defend against unauthorized access and data breaches. 2. **Evaluate Authentication Frameworks:** Compare various methods for **Key Distribution** and **User Authentication**, identifying the most secure protocols for different network environments. 3. **Appraise Cloud and Wireless Security:** Critically assess the unique security risks associated with **Cloud Computing** and **Wireless Networks**, including

	the implementation of effective Access Control Lists (ACLs). 4. **Deconstruct Security Protocols:** Explain the inner workings of **Transport-Level Security (TLS)** and **IP Security (IPsec)**, detailing how they provide end-to-end encryption and data integrity. 5. **Distinguish Multimedia Protection Techniques:** Differentiate between **Steganography** (hiding information) and **Digital Watermarking** (protecting ownership) across various media types like image, audio, and video. 6. **Formulate Media Security Strategies:** Design theoretical models for embedding and extracting watermarks to ensure copyright protection and content authentication.
Indicative Contents المحتويات الإرشادية	**Foundations and Identity Management** • **Security Architecture:** The OSI Security Architecture, security attacks (passive vs. active), and security services/mechanisms. • **Key Distribution:** Symmetric key distribution using Key Distribution Centers (KDC); Asymmetric key distribution and Public Key Infrastructure (PKI). • **Authentication Protocols:** Kerberos, X.509 Directory Authentication service, and federated identity management. **Access Control and Modern Infrastructure** • **Network Access Control (NAC):** Elements of a NAC system, EAP (Extensible Authentication Protocol), and IEEE 802.1X port-based authentication. • **Cloud Security:** Cloud computing service models (SaaS, PaaS, IaaS); Cloud security risks and countermeasures; Data protection in the cloud. **Network Layer & Transport Security** • **Transport-Level Security:** SSL/TLS architecture, TLS Handshake Protocol, Heartbeat Protocol, and HTTPS. • **Wireless Security:** IEEE 802.11i wireless LAN security, WPA2/WPA3 protocols, and common wireless threats (Evil Twin, War Driving). • **IP Security (IPsec):** IPsec documents, Encapsulating Security Payload (ESP), Authentication Header (AH), and Tunnel vs. Transport modes. **Multimedia Concepts and Steganography** • **Digital Media Fundamentals:** Characteristics of digital images, audio, and video; Redundancy and human perception (HVS/HAS). • **Steganography:** The prisoner's problem; Spatial domain techniques (Least Significant Bit - LSB); Transform domain techniques (DCT/DWT basics); Steganalysis (detecting hidden messages). **Digital Watermarking and Copyright** • **Watermarking Principles:** Requirements (Invisibility, Robustness, Capacity); Fragile vs. Robust watermarks.

	• **Application-Specific Watermarking:**
	  ○ **Image Watermarking:** Embedding marks in different color spaces (RGB vs. YCbCr).
	  ○ **Audio Watermarking:** Echo hiding, phase coding, and spread spectrum.
	  ○ **Video Watermarking:** Motion vector watermarking and inter-frame/intra-frame embedding.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	**Teaching Strategies:** 1- Interactive Lectures: Incorporate multimedia elements into lectures to demonstrate concepts visually and engage students effectively. 2- Hands-on Labs: Provide hands-on lab exercises where students can practice implementing security measures in multimedia networks. 3- Group Projects: Assign group projects that require students to analyze and propose solutions for multimedia security issues in network environments. 4- Peer Teaching: Encourage students to teach each other about specific topics related to multimedia security and networks. **A- Learning Strategies:** 1- Active Learning: Encourage students to actively engage with the material through discussions, hands-on activities, and problem-solving tasks. 2- Visual Aids: Use multimedia presentations, videos, and diagrams to help students visualize complex concepts. 3- Collaborative Learning: Foster collaboration among students through group projects and discussions on multimedia security topics. 4- Practice Exercises: Offer practice exercises and quizzes to reinforce learning and assess understanding.

	5- Real-world Applications: Connect theoretical concepts to real-world applications of multimedia security and networks to demonstrate their relevance. 6- Feedback Mechanisms: Provide timely feedback on assignments and assessments to help students track their progress and improve their understanding. 7- Continuous Assessment: Implement regular assessments to gauge student comprehension and adjust teaching strategies as needed.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	5-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

Week No.	Material Covered

Week 1	**Introduction to Network security**
Week 2	**Key Distribution & User Authentication**
Week 3	**Network Access Control**
Week 4	**Cloud Security**
Week 5	**Transport-Level Security**
Week 6	**Transport-Level Security(cont.)**
Week 7	**Wireless Network Security**
Week 8	**Wireless Network Security(cont.)**
Week 9	**IP Security**
Week 10	**IP Security(cont.)**
Week 11	**Introduction of Multimedia**
Week 12	**Multimedia Security Concepts**
Week 13	**Steganography Concepts**
Week 14	**Digital Watermarking**
Week 15	**Digital Watermarking(cont.)**

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	**Material Covered**
Week 1	Introductory Lecture to python
Week 2	Network Reconnaissance
Week 3	Key Exchange
Week 4	Authentication
Week 5	Access Control Lists
Week 6	Analyzing TLS
Week 7	Exam
Week 8	Introduction to Steganography
Week 9	Steganography-Embedding Grey Image in color image
Week 10	Steganography-Embedding Grey Image in color image (Cont.)
Week 11	Introduction to Digital Watermarking

Week 12	Digital Watermarking (Same size)
Week 13	Digital Watermarking (Different size)
Week 14	Audio Watermarking
Week 15	Video Watermarking

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Stallings, William. "Network security essentials: applications and standards." (2014). Shih F.Y. (Ed.) - Multimedia Security: Watermarking, Steganography, and Forensics 2012.	
Recommended Texts		
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	Data Mining			**Module Delivery**	
Module Type	Core			☐ **Theory** ☒ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code					
ECTS Credits	8				
SWL (hr/sem)	200				
Module Level		1	**Semester of Delivery**		1st
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Hasan Naji Ali		**e-mail**	hasan@tu.edu.iq	
Module Leader's Acad. Title		Assistant Lecturer	**Module Leader's Qualification**		Msc
Peer Reviewer Name		Mahammed Aktham	**e-mail**		
Scientific Committee Approval Date		1/2/2026	**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	None	**Semester**	
Co-requisites module	Understanding basics of Data Mining	**Semester**	2

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	• Introduce students to the concept of Data Mining and its importance in decision support. • Explain the Knowledge Discovery in Databases (KDD) process. • Enable students to use classification algorithms. • Study clustering algorithms. • Understand association rules. • Analyze time-series data and forecasting. • Train students to use practical tools such as Python and RapidMiner. • Prepare students for the job market in data analysis and artificial intelligence.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	• Explain fundamental Data Mining concepts and distinguish them from Machine Learning and Statistics. • Apply classification algorithms such as KNN, Decision Tree, and Naïve Bayes. • Apply clustering algorithms such as K-means. • Extract association rules using the Apriori algorithm. • Evaluate model performance using Accuracy, Precision, and Recall. • Preprocess and clean data before analysis. • Implement real-world data analysis projects using Python. • Present professional analytical reports.
Indicative Contents المحتويات الإرشادية	• Introduction to Data Mining and its relation to KDD. • Data Preprocessing. • Classification. • Decision Trees. • K-Nearest Neighbors algorithm. • Clustering. • Association Rules. • Model Evaluation. • Data Mining applications in banking, marketing, cybersecurity, and healthcare.

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	• Theoretical lectures explaining concepts and algorithms. • Interactive discussions on real-world applications. • Practical lab sessions using Python (Pandas, Scikit-learn, Matplotlib). • Analysis of real datasets such as Iris, Titanic, and Market Basket datasets. • Group projects analyzing real datasets.

	• Preparation and presentation of analytical reports.

Student Workload (SWL)
الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	93	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	6.2
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	107	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	7.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	200		

Module Evaluation
تقييم المادة الدراسية

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

Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	Concept of Data
Week 2	Introduction to Data Mining
Week 3	KDD Process
Week 4	Data Preprocessing
Week 5	Data Cleaning & Transformation
Week 6	Classification Overview
Week 7	Decision Tree

Week 8	KNN
Week 9	Midterm Review
Week 10	Naïve Bayes
Week 11	Association Rules
Week 12	Apriori Algorithm
Week 13	Model Evaluation
Week 14	Data Warehouse
Week 15	Reviewing Students' Projects

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	
Week 1	Install Python and Jupyter
Week 2	Using Pandas
Week 3	Reading Datasets
Week 4	Data Processing
Week 5	Normalization
Week 6	Apply KNN
Week 7	Model Evaluation
Week 8	K-means
Week 9	Association Rules
Week 10	Apply Decision Tree

Week 11-14	Practical Project
Week 15	Project Presentations

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	*Abraham* Silberschatz (2022). Operating System Concept nine edition.	Yes
Recommended Texts	Introduction to Data Mining – Pang-Ning Tan	
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	Neural Networks			**Module Delivery**	
Module Type	Core			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code					
ECTS Credits					
SWL (hr/sem)	148				
Module Level		4	**Semester of Delivery**		2nd
Administering Department		Computer Science	**College**	CCSM	
Module Leader	Salwa Khalid Abdulateef		**e-mail**	Khalid.salwa@tu.edu.iq	
Module Leader's Acad. Title		Asst. Prof.	**Module Leader's Qualification**		MSc.
Module Tutor			**e-mail**		
Peer Reviewer Name		Mohamed Aktham	**e-mail**		
Scientific Committee Approval Date		02/01/2026	**Version Number**		1.0

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Machine learning, Artificial intelligent	**Semester**	
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	19. learning the types of neural networks. 20. Knowing how these neural networks work, especially by linking them to practical life. 21. Knowing the problems that appear in neural networks and ways to solve them. 22. The student can be ability to write programs for neural networks. 23. The ability to use neural networks in some research 24. Learning how to choose the best network for your algorithm. 25. A detailed explanation of algorithms and how to program them. 26. Teaching the student to calculate the complexity time of the algorithm. 27. Encouraging the student to be creative and think about the specialization projects and keep pace with the development taking place in this field.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	5. The student is supposed to understand how to start building simple algorithms and know the language of building algorithms (pseudo code). 6. Build basic causal skills in creating and proving the correctness of algorithms and programs. 7. Build skills in analyzing data and acquiring knowledge 8. Build skills in how to choose appropriate solutions for some problems and assign the best algorithms to solve them 9. Enabling students to think and analyze mathematical topics. 10. Enabling students to think and analyze topics related to algorithms. 11. Giving the learner the skills to be able to build a neural network
Indicative Contents المحتويات الإرشادية	1. Review of Basic Programming Concepts: - Cv2, Pandas, Numpy, Keras, and Matplot. 2. overview on Fundamentals of artificial neural networks Theory of Neural Networks (NN) Neural Networks Applications and properties 3. Types of activation functions such as sigmoid, relu and others 4. Models of Neuron 5. Logical Operations: such as AND, OR, and XOR. 6. Types of learning Supervised, unsupervised 7.Network Architectures Signal layer and Multiple Layers of Neurons, 8.Perceptron and Recurrent (Feedback) Networks , Learning algorithm

	9. Explain Backpropagation Neural Networks algorithms and programming. 10. Explain Hopfield Network algorithms mathematical and programming. -

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	1. Lectures: The instructor will deliver lectures to introduce and explain neural network concepts, and problem-solving techniques. This will provide students with a solid theoretical foundation. 2. Interactive Discussions: Engaging students in interactive discussions allows them to ask questions, seek clarifications, and participate actively in the learning process. Discussions can include reviewing code examples, discussing programming best practices, and exploring real-world applications of programming concepts. 3. Laboratory Sessions: Laboratory sessions are dedicated practical sessions where students apply the concepts learned in lectures to hands-on programming exercises. Key strategies for the laboratory sessions include: a. Programming Exercises: Students will work on programming exercises and projects in the laboratory, providing them with practical experience in coding and problem-solving. b. Guided Practice: Lab instructors or teaching assistants will be available to provide guidance, assistance, and immediate feedback on students' code. They can help students debug their programs, identify errors, and improve their coding skills. c. Collaboration and Peer Learning: Students can collaborate with their peers in the laboratory, fostering teamwork and enabling knowledge sharing. Working together on programming tasks promotes discussions, problem-solving, and peer learning. 4. Code Reviews and Feedback: The instructor will provide feedback on

	students' code, reviewing their solutions, and offering suggestions for improvement. This feedback will help students enhance their coding skills and adhere to best practices.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	56	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	3.7
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	92	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	6.13
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	148		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5, 11	
	Assignments	4	20% (20)	7, 12	
	Projects	1	20% (20)	5-14	
	Report	1			
Summative assessment	**Midterm Exam**	2 hr	10% (10)	11	
	Final Exam	2hr	40% (40)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

Week No.	Material Covered
Week 1	**History and Fundamentals of artificial neural networks**
Week 2	**Theory of Neural Networks (NN)**
Week 3	**Neural Networks Applications and properties**
Week 4	**Types of activation functions**
Week 5	**Exam1**
Week 6	**Models of Neuron**

Week 7	**Logical Operations**
Week 8	**Types of learning**
Week 9	**Exam2**
Week 10	**Network Architectures**
Week 11	**Multiple Layers of Neurons,**
Week 12	**Perceptron and Recurrent (Feedback) Networks , Learning algorithm**
Week 13	**Backpropagation Neural Networks**
Week 14	**Backpropagation Neural Networks**
Week 15	**Hopfield Network**

Delivery Plan (Weekly Lab. Syllabus): المنهاج الاسبوعي للمختبر:	
Week No.	Material Covered
Week 1	Explaining the basics of programming in python
Week 2	Explaining the libraries in python
Week 3	Activation functions
Week 4	Activation functions
Week 5	Neural Network Toolbox
Week 6	Weights
Week 7	How can build a simple neural network
Week 8	can build a single layer neural network
Week 9	can build a multi layers neural network

Week 10	perceptron algorithm use to solve a simple classification logical-AND problem.
Week 11	perceptron algorithm use to solve a simple classification logical-OR problem.
Week 12	perceptron algorithm use to solve a simple classification logical- XOR problem.
Week 13	Back propagation learning algorithm
Week 14	Hopfield learning algorithm
Week 15	Examples

Learning and Teaching Resources

مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts		
Recommended Texts	**1** Deep Learning for Vision Systems by MOHAMED ELGENDY: Manning Publications, 2020 . **2** Neural Networks: Tricks of the Trade by G. Montavon, G. Orr, K.R. Muller, (Springer, New York, 2012 **3** Neural Networks and Deep Learning: by Charu C. Aggarwal, 2018 textbook **4** An Introduction to Machine Learning, by Miroslav¬Kubat , Springer International Publishing AG 2017, DOI 10.1007/978-3-319-63913-0.	
Websites	• https://www.ibm.com/topics/neural-networks. • https://en.wikipedia.org/wiki/Neural_network	

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