Ministry of Higher Education and Scientific Research
Scientific Supervision and Scientific Evaluation Apparatus
Directorate of Quality Assurance and Academic Accreditation
Accreditation Department

Academic Program and Course Description Guide

2024/2025

Introduction:

The educational program is a well-planned set of courses that include procedures and experiences arranged in the form of an academic syllabus. Its main goal is to improve and build graduates' skills so they are ready for the job market. The program is reviewed and evaluated every year through internal or external audit procedures and programs like the External Examiner Program.

The academic program description is a short summary of the main features of the program and its courses. It shows what skills students are working to develop based on the program's goals. This description is very important because it is the main part of getting the program accredited, and it is written by the teaching staff together under the supervision of scientific committees in the scientific departments.

This guide, in its second version, includes a description of the academic program after updating the subjects and paragraphs of the previous guide in light of the updates and developments of the educational system in Iraq, which included the description of the academic program in its traditional form (annual, quarterly), as well as the adoption of the academic program description circulated according to the letter of the Department of Studies T 3/2906 on 3/5/2023 regarding the programs that adopt the Bologna Process as the basis for their work.

In this regard, we can only emphasize the importance of writing an academic programs and course description to ensure the proper functioning of the educational process.

Concepts and terminology:

Academic Program Description: The academic program description provides a brief summary of its vision, mission and objectives, including an accurate description of the targeted learning outcomes according to specific learning strategies.

Course Description: Provides a brief summary of the most important characteristics of the course and the learning outcomes expected of the students to achieve, proving whether they have made the most of the available learning opportunities. It is derived from the program description.

Program Vision: An ambitious picture for the future of the academic program to be sophisticated, inspiring, stimulating, realistic and applicable.

Program Mission: Briefly outlines the objectives and activities necessary to achieve them and defines the program's development paths and directions.

Program Objectives: They are statements that describe what the academic program intends to achieve within a specific period of time and are measurable and observable.

Curriculum Structure: All courses / subjects included in the academic program according to the approved learning system (quarterly, annual, Bologna Process) whether it is a requirement (ministry, university, college and scientific department) with the number of credit hours.

Learning Outcomes: A compatible set of knowledge, skills and values acquired by students after the successful completion of the academic program and must determine the learning outcomes of each course in a way that achieves the objectives of the program.

Teaching and learning strategies: They are the strategies used by the faculty members to develop students' teaching and learning, and they are plans that are followed to reach the learning goals. They describe all classroom and extra-curricular activities to achieve the learning outcomes of the program.

Academic Program Description Form

University Name: .....Diyala.....
Faculty/Institute: .....College of Sciences.....
Scientific Department: .....Computer Sciences.....
Academic or Professional Program Name: .....Bachelor's in Computer Sciences.....
Final Certificate Name: ..... Bachelor's in Computer Sciences.....
Academic System: …Semester Study System……
Description Preparation Date: 10/9/2024
File Completion Date: 10/9/2024

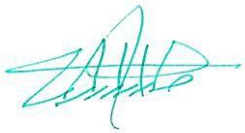
Signature:
Head of Department Name: Bashar Talib Hemeed

Date: 12/9/2024

Signature::
Scientific Associate Name: Munther Hamza Radi

Date:12/9/2024

The file is checked by:
Department of Quality Assurance and University Performance
Director of the Quality Assurance and University Performance Department: Ghassan Sabeeh Mahmood
Date: 12/9/2024
Signature:



Approval of the Dean
Prof: Taha Muhammad Hassan

1. Program Vision

Computer science is considered one of the modern applied sciences intertwined with all other sciences. It occupied a prominent degree and position in scientific studies in colleges and universities in the world. The establishment of the Computer Science Department in the College of Science at Diyala University came with the establishment of the college in the year 2002-2003, and since that date I graduated from the department. The sixteenth session, the last of which was the 2020-2021 session, and postgraduate studies were introduced in the year 2015-2016 to obtain a master's degree, and 20 students were accepted for the academic year 2020-2021. In the same year, a doctoral study was introduced in computer science, and it is hoped that doctoral students will be accepted for the academic year 2021 -2022.

2. Program Mission

The Computer academic staff pursues a multifaceted charge at (Diyala) University. The Department of Computer Science, since its inception, has enjoyed a distinguished scientific reputation among the faculties of science. In general, the department dedicates the capabilities to maintain the highest scientific level, as the department provides a wide study and research program within the disciplines of computer science. This is to provide academic opportunities for undergraduate and postgraduate students. Every year, many original scientific researches are carried out in various branches of computer science. At the level of preliminary studies, the department provides the opportunity to obtain a bachelor's degree in computer science by giving practical and theoretical lessons, as well as scientific research in all the specializations of the department, as the subjects taught during the four academic years qualify graduates to work according to their specialization, wherever they are needed in the fields of the jobs.

3. Program Objectives

The objective of the service courses is to teach specific programming languages, operating systems, environments, and other specific knowledge. They are to serve the community, other programs within the University, and majors and minors, by teaching application-area-specific knowledge to students.

The overall objective of the Computer Science faculty is to successfully implement the Computer Science major. In order to do this, the faculty have the objectives of successfully teaching a variety of courses, using current technology, giving students hands-on experiences, renewing their professional skills and knowledge base, sharing activities and knowledge with other professionals, and regularly reviewing and updating curriculum based on professional organization guidelines and both student and advisory board input.

4. Program Accreditation

Does the program have program accreditation? And from which agency? NO

5. Other external influences

Is there a sponsor for the program? NO

6. Program Structure

Program Structure	Number of Courses	Credit hours	Percentage	Reviews*
Institution Requirements	**4**	**4**		
College Requirements	**----**	**---**		
Department Requirements	**54**	**162**		
Summer Training				
Other				

* This can include notes whether the course is basic or optional.

7. Program Description

Bologna Process

Frist stage

Year/Level	Course Code	Course Name	Credit Hours	
			theoretical	practical
Semester 1	**COS-101**	**Differentiation Methods**	**3**	**---**
	COM-111	**Introduction to Programming**	**2**	**2**

Bologna Process	COM-112	Computational mathematics	3	---
	COM-113	Computer Skills (I)	2	2
	COM-114	Digital Logic	2	2
	UD11	English Language	2	---
Semester 2	COS-102	Integration Methods	3	---
	COM-121	Programming Fundamentals	2	2
Bologna Process	COM-122	Discrete Structures	3	---
	COM-123	Computer Organization	2	2
	UD14	Human Rights and democracy	2	---
	UD12	Arabic Language	2	---
Second stage				
Semester 3	COM-211	Introduction to Object Oriented Language	2	2
	COM-212	Data Structures	2	2
Bologna Process	COM-213	Computation Theory	3	---
	COM-214	Web Design and Programming	2	2
	SCI-103	Numerical Methods	2	2
	UD24	Baath Party Crimes in Iraq	2	---
Semester 4	COM-221	Algorithms Design and Analysis	2	2
	COM-222	Computer Graphics	2	2
Bologna Process	COM-223	Computer Architecture	3	---
	COM-224	Introduction to Python	2	2
	COM-225	Object Oriented Programming	2	2
	COM-225	Visual Programming	2	2
	UD21	English Language 2	2	-
	UD22	Arabic Language	2	-
Third stage				
Third stage	COM-311	Language Translator	2	2
	COM-312	Artificial Intelligence	2	2
Frist Course	COM-313	Cryptography	2	2
	COM-314	Digital Image Processing	2	2
	COM-315	Introduction to Database	2	2
	COM-316	Research Methodology	3	---
	COM-321	Software Engineering	2	2

Third stage **Second Course**	**COM-322**	**Web applications Development**	**2**	**2**
	COM-323	**Computer Security**	**2**	**2**
	COM-324	**Knowledge Representation**	**3**	**---**
	COM-325	**Distributed Database**	**2**	**2**
	COM-326	**Pattern Recognition**	**2**	**2**
Fourth stage				
Fourth stage **Frist Course**	**COM-411**	**Introduction Operating Systems**	**2**	**2**
	COM-412	**Machine Learning**	**2**	**2**
	COM-413	**Computer Networks**	**2**	**2**
	COM-414	**Multimedia**	**2**	**2**
	COM-415	**Coding and Data Compression**	**3**	**---**
	SCI-106	**Research Project (I)**	**---**	**6**
Fourth stage **Second Course**	**COM-421**	**Techniques of Operating Systems**	**2**	**2**
	COM-422	**Data Mining**	**3**	**---**
	COM-423	**Network Security**	**3**	**---**
	COM-424	**Modeling and Simulation**	**3**	**---**
	COM-425	**Evolutionary Computing**	**3**	**---**
	SCI-107	**Research Project (II)**	**---**	**5**

8. Expected learning outcomes of the program

Knowledge
A1:: design and represent the flow of simple computer programs in a standard design language;
A2:: describe and use the basic concepts of classes and objects in computer programs;
A3:: create and manipulate simple databases;
A4:: describe the structure of a computing system, the design of its basic components and explain the interactions of hardware and software components;
Skills
B1:: manipulate data and data representation through logical and numerical techniques;
B2:: understand fundamental concepts of network tools;
B3:: describe the file architecture and the organization of a web site;
B4:: describe the main ethical, social, legal and professional issues in Computer Science and Software Engineering;
Ethics
C1:: Creating a spirit of teamwork among students through laboratory groups, graduation projects, and reports.
C2:: Boosting students' self-confidence through daily discussions via seminars.

C3:: Enhancing students' confidence in facing challenges by completing their academic journey through acquiring skills to engage in the job market.
C4:: Statement of the positive effects when embodying these values, and the negative effects when lacking them, through encouragement, intimidation, intrigue, and motivation.

9. Teaching and Learning Strategies

- Explanation using various modern presentation tools:
- Lecture method and use of interactive whiteboard.
- Providing students with basics and additional topics related to software programs.
- Forming discussion groups during lectures to discuss modern systems that require thinking and analysis.
- Asking students a set of critical thinking questions during lectures, such as what, how, when, and why, about specific topics.
- Assigning students homework that requires self-explanatory causal explanations.

10. Evaluation methods

- Practical exams
- Theoretical exams
- Reports and studies
- Daily quizzes with self-solvable questions
- Grades determined by homework assignments

11.Faculty

Faculty Members

Academic Rank	Specialization		Special Requirements/Skills (if applicable)		Number of the teaching staff	
	General	Special			Staff	Lecturer
Professor	Computer Science	Artificial Intelligence Simulation			Staff	
Professor	Computer Science	Computer Science			Staff	

Professor	Computer Science	Technology and Computer Science			**Staff**	
Professor	Computer Science	Object-Oriented and Visual Programming			**Staff**	
Professor	Computer Science	Computer Application Technologies			**Staff**	
Professor	Computer Science	Artificial Intelligence			**Staff**	
Professor	Computer Science	Information Technology			**Staff**	
Assistant Professor	Computer Science	Bioinformatics			**Staff**	
Assistant Professor	Computer Science	Information Technology			**Staff**	
Assistant Professor	Computer Science	Computing/Information Technology			**Staff**	
Assistant Professor	Computer Science	Computer Networks			**Staff**	
Assistant Professor	Computer Science	Information Technology			**Staff**	
Assistant Professor	Computer Science	Information Technology			**Staff**	
Assistant Professor	Electrical Engineering	Communications			**Staff**	
Assistant Professor	Computer Science	Computer Science/Cyber security			**Staff**	
Assistant Professor	Computer Science	Computer Science/Cyber security			**Staff**	
Assistant Professor	Computer Science	Computer Science/Cyber security/Data Encryption			**Staff**	
Assistant Professor	Computer Science	Artificial Intelligence			**Staff**	
Assistant Professor	Computer Science	Artificial Intelligence			**Staff**	

Lecturer	Computer Science	Computer Science			**Staff**	
Lecturer	Computer Science	Computer Science			**Staff**	
Lecturer	Computer Science	Computer Science			**Staff**	
Lecturer	Computer Science	Computer Science			**Staff**	
Assistant Lecturer	Computer Science	Computer Science			**Staff**	
Assistant Lecturer	Computer Science	Computer Science			**Staff**	

Professional Development

Mentoring new faculty members

1. Conducting developmental courses for newly hired individuals to enhance their competencies.
2. Hosting developmental seminars for newly hired individuals to enhance their competencies.
3. Organizing developmental workshops for newly hired individuals to enhance their competencies.

Professional development of faculty members

Participating in delivering lectures for secondary subjects under the supervision of the department chair to prepare for giving lectures on core subjects.

12. Acceptance Criterion

central admission

13. The most important sources of information about the program

Ministry of Higher Education and Scientific Research
University of Diyala
College of Science
Department of Computer Science
College of Science website https://sciences.uodiyala.edu.iq/

14. Program Development Plan
We are seeking academic program accreditation

Program Skills Outline															
				Required program Learning outcomes											
Year/Level	Course Code	Course Name	Basic or optional	Knowledge				Skills				Ethics			
				A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4
Semester 1	COS-101	Differentiation Methods	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-111	Introduction to Programming	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-112	Computational mathematics	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-113	Computer Skills (I)	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-114	Digital Logic	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	UD11	English Language	optional	●	●	●	●	●	●	●	●	●	●	●	●

Semester 2	**COS-102**	**Integration Methods**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-121	**Programming Fundamentals**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-122	**Discrete Structures**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-123	**Computer Organization**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	UD14	**Human Rights and democracy**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	UD12	**Arabic Language**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
Semester 3	**COM-211**	**Introduction to Object Oriented Language**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-212	**Data Structures**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-213	**Computation Theory**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-214	**Web Design and Programming**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	SCI-103	**Numerical Methods**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●

	UD24	Baath Party Crimes in Iraq	Basic	●	●	●	●	●	●	●	●	●	●	●	●
Semester 4	COM-221	Algorithms Design and Analysis	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-222	Computer Graphics	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-223	Computer Architecture	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-224	Introduction to Python	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-225	Object Oriented Programming	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-225	Visual Programming	Basic	●	●	●	●	●	●	●	●	●	●	●	●
Semester 5	COM-311	Language Translator	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-312	Artificial Intelligence	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-313	Cryptography	Basic	●	●	●	●	●	●	●	●	●	●	●	●
	COM-314	Digital Image Processing	Basic	●	●	●	●	●	●	●	●	●	●	●	●

	COM-315	**Introduction to Database**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-316	**Research Methodology**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
Semester 6	**COM-321**	**Software Engineering**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-322	**Web applications Development**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-323	**Computer Security**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-324	**Knowledge Representation**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-325	**Distributed Database**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-326	**Pattern Recognition**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
Semester 7	**COM-411**	**Introduction Operating Systems**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-412	**Machine Learning**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-413	**Computer Networks**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●

	COM-414	**Multimedia**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-415	**Coding and Data Compression**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	SCI-106	**Research Project (I)**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
Semester 8	**COM-421**	**Techniques of Operating Systems**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-422	**Data Mining**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-423	**Network Security**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-424	**Modeling and Simulation**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	COM-425	**Evolutionary Computing**	**optional**	●	●	●	●	●	●	●	●	●	●	●	●	●
	SCI-107	**Research Project (II)**	**Basic**	●	●	●	●	●	●	●	●	●	●	●	●	●

- **Please tick the boxes corresponding to the individual program learning outcomes under evaluation.**

1. Course Name: **Coding and Data Compression**

2. Course Code: **COM-415**

3. Semester / Year: Semester 2024/2025

4. Description Preparation Date: 1/9/2024

5. Available Attendance Forms: Mandatory attendance

6. Number of Credit Hours (Total) / Number of Units (Total) : 3 hours theoretical + per week / 3 Units

7. Course administrator's name (mention all, if more than one name)

Name: Dr. Burhan Molan saleh
Email:

8. Course Objectives

The essential course "Data Encoding and Compression" focuses on teaching students the key elements of data encoding and compression scientifically and specifically. This includes the types of algorithms used in data encoding and compression, their differences, advantages, disadvantages, and implications. The curriculum aims to:

1- Ensure that students understand the principles of data encoding and compression, in addition to the applied fields in which this field is involved, in order to successfully meet the course requirements.

2- Familiarize students with various research methods and types.

3- Enhance students' ability to use available software in this field, along with the programming and compression skills they acquire.

.....
.....
.....

9. Teaching and Learning Strategies

Strategy	1- Enable students to solve problems related to the intellectual framework of the scientific research method.

	2- Enable students to solve problems in a scientific manner and purely scientific bases

10. Course Structure

Evaluation method	Education method	Unit name and/or topic	Required learning outcomes	hours	Week
Oral or written test	Electronic lecture using Microsoft Editor	Introduction to coding and data compression		4	**1**
Oral or written test	Electronic lecture using Microsoft Editor	Basic techniques and methods		4	**2**
Oral or written test	Electronic lecture using Microsoft Editor	**Basic VL codes**		4	**3**
Oral or written test	Electronic lecture using Microsoft Editor	**Statistical Methods**		4	**4**
Oral or written test	Electronic lecture using Microsoft Editor	**Statistical Methods**		4	**5**
Oral or written test	Electronic lecture using Microsoft Editor	**Image compression**		4	**6**
Oral or written	Electronic lecture using	**Image compression**		4	**7**
		Exam1		4	**8**
	Electronic	**Wavelet methods**		4	**9**
		Video compression		4	**10**
Oral or written test	Electronic lecture using	**Video**		4	**11**

	Microsoft Editor	**compression**			
Oral or written test	Electronic lecture using Microsoft Editor	**Audio compression**		4	**12**
Oral or written test	Electronic lecture using Microsoft Editor	**Audio compression**		4	13
Oral or written test	Electronic lecture using Microsoft Editor	**Presentation**		4	14
Oral or written test	Electronic lecture using Microsoft Editor	**Exam2**		4	15
				4	16

11.Course Evaluation

Practical exams
2- Theoretical exams
3- Reports and studies
4- Daily quizzes with self-solvable questions
5- Grades determined by homework assignments

12.Learning and Teaching Resources

The required textbooks (if any) and methodology	**Handbook of Data Compression fifth**
The main references (sources):	**Sayood, Khalid. *Introduction to data compression*. Morgan Kaufmann, 2017.**
Recommended supporting books and references (scientific journals, reports, etc.):	***Salomon, David, and Giovanni Motta. Handbook of data compression. London; New York: Springer,, 2010.***

Course Description Form

13. Course Name:
Computational Security1
14. Course Code:
COM23
15. Semester / Year: 2024/2025
Semester
16. Description Preparation Date:
1/9/2024
17. Available Attendance Forms:
Attendance inside the university
18. Number of Credit Hours (Total) / Number of Units (Total) **3**
2 theoretical + 2 practical sessions per week.
19. Course administrator's name (mention all, if more than one name)
Name: *Ghasan mahmod sabih* Email:
20. Course Objectives

Course Objectives	• **The aim is to prepare scientifically minded students who can recognize characteristics of efficient programs and utilize data protection features addition to what is used in programming languages.**
21. Teaching and Learning Strategies	
Strategy	• Explaining using various modern presentation tools: Lecture method and using interac whiteboard. • Providing students with basics and additional topics related to data protection. • Forming discussion groups during lectures to discuss modern systems that req thinking and analysis. • Asking students a set of critical thinking questions during lectures such as what, h when, and why for specific topics. • Assigning students homework that requires self-explanatory causal explanations.

22. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1	3	Introduction		Lecture Base	
2	3	Cryptography		Lecture Base	
3	3	Stream Ciphers		Lecture Base	
4	3	Data Encryption Standard (DES) part1		Lecture Base	
5	3	Data Encryption Standard (DES) part2		Lecture Base	
6	3	Advanced Encryption Standard (AES) part 1		Lecture Base	
7	3	Advanced Encryption Standard (AES) part 2		Lecture Base	
8	3	Exam 1		Lecture Base	
9	3	More About Block Ciphers		Lecture Base	
10	3	Introduction to Public-Key Cryptography		Lecture Base	
11	3	Essential Number Theory for Public-Key Algorithms		Lecture Base	
12	3	The RSA Cryptosystem		Lecture Base	
13	3	Finding Large Primes for RSA , Attacks and countermeasures		Lecture Base	
14	3	Public-Key Cryptosystems Based on the Discrete Logarithm Problem		Lecture Base	
15	2	Exam 2		Lecture Base	

23.Course Evaluation

Practical exams

2- Theoretical exams

3- Reports and studies

4- Daily quizzes with self-solvable questions

5- Grades determined by homework assignments

24.Learning and Teaching Resources

Required textbooks (curricular books, if any)	

Main references (sources)	Understanding Cryptography, Christof Paar · Jan Pelzl . • Cryptography and Network Security: Principles and Practice, William Stallings
Electronic References, Websites	

Course Description Form

25. Course Name:
Computational Security2
26. Course Code:
COM24
27. Semester / Year: 2024/2025
Semester
28. Description Preparation Date:
1/9/2024
29. Available Attendance Forms:
Attendance inside the university
30. Number of Credit Hours (Total) / Number of Units (Total) **3**
2 theoretical + 2 practical sessions per week.
31. Course administrator's name (mention all, if more than one name)
Name: *Ghasan mahmod sabih* Email:
32. Course Objectives

Course Objectives	• **The aim is to prepare scientifically minded students who can recognize characteristics of efficient programs and utilize data protection features addition to what is used in programming languages.**
33. Teaching and Learning Strategies	
Strategy	• Explaining using various modern presentation tools: Lecture method and using interac whiteboard. • Providing students with basics and additional topics related to data protection. • Forming discussion groups during lectures to discuss modern systems that req thinking and analysis. • Asking students a set of critical thinking questions during lectures such as what, h when, and why for specific topics. • Assigning students homework that requires self-explanatory causal explanations.

34. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1	3	Introduction		Lecture Base	
2	3	Elliptic Curve Cryptosystems		Lecture Base	
3	3	Elliptic Curve Models		Lecture Base	
4	3	Digital Signatures Digital Signatures Models		Lecture Base	
5	3			Lecture Base	
6	3	Hash Functions Hash Functions Family		Lecture Base	
7	3	Exam 1		Lecture Base	
8	3	Message Authentication Codes (MACs)		Lecture Base	
9	3	Key Establishment		Lecture Base	
10	3	Access controls		Lecture Base	
11	3	Models of Access controls		Lecture Base	
12	3	Introduction		Lecture Base	
13	3	Protocols		Lecture Base	
14	3	Protocols types		Lecture Base	
15	2	Exam 2		Lecture Base	

35.Course Evaluation

Practical exams

2- Theoretical exams

3- Reports and studies

4- Daily quizzes with self-solvable questions

5- Grades determined by homework assignments

36.Learning and Teaching Resources

Required textbooks (curricular books, if any)	
Main references (sources)	Understanding Cryptography, Christof Paar · Jan Pelzl . • Cryptography and Network Security: Principles and Practice, William Stallings
Electronic References, Websites	

Course Description Form

37. Course Name: Evolutionary Computing

38. Course Code: COM-425

39. Semester / Year: Second Semester / 2024-2025

40. Description Preparation Date: 1/9/2024

41. Available Attendance Forms: Mandatory attendance

42. Number of Credit Hours (Total) / 3 hours theoretical per week / 3 Units

43. Course administrator's name (mention all, if more than one name)

Name: Prof.Dr. Ziyad Tariq Mustafa Al-Ta'i
Email: Ziyad1964tariq@uodiyala.edu.iq

44. Course Objectives

Teaching the student machine learning methods (the calculator) that depend on inference and prediction of future events through statistical methods, artificial networks and their practical (software) applications aimed at the development of technological innovations.
Teaching and educating students on all the necessary and necessary information related to mach learning, which qualifies them to work and research in all areas of research and prediction methods

45. Teaching and Learning Strategies

Strategy	1- Enable students to solve problems related to the intellectual framework of the scientific research method. 2- Enable students to solve problems in a scientific manner and on pur scientific bases

10. Course Structure

Evaluation method	Education method	Unit name and/or topic	Required learning outcomes	hours	Week
Oral or written test	Electronic lecture using Microsoft Editor	Evolutionary Computation Introduction		3	**1**
Oral or written test	Electronic lecture using Microsoft Editor	Biological Evolutionary theory		3	**2**
Oral or written test	Electronic lecture using Microsoft Editor	Genetic Algorithms Types		3	**3**
Oral or written test	Electronic lecture using Microsoft Editor	Seeding the population and Encoding		3	**4**
Oral or written test	Electronic lecture using Microsoft Editor	Selection		3	**5**
Oral or written test	Electronic lecture using Microsoft Editor	Selection Types		3	**6**
Oral or written	Electronic	Crossover		3	**7**
				3	**8**
	Electronic	Mutation		3	**9**
				3	**10**
Oral or written test	Electronic lecture using Microsoft Editor	Complete Example		3	**11**
Oral or written test	Electronic lecture using Microsoft Editor	TSP using Genetic		3	**12**
Oral or written test	Electronic lecture using Microsoft Editor	Routing using genetic		3	13
Oral or written	Electronic	4 Queen puzzle using genetic		3	14

test	lecture using Microsoft Editor				
Oral or written test	Electronic lecture using Microsoft Editor	Genetic tutorial		3	15

11. Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, dailyoral, monthly, or written exams, reports .... etc

12. Learning and Teaching Resources

Required textbo (curricular book any)	Machine Learning, Tom Mitchell, McGraw Hill Press, 1997
Main referen (sources)	1- Fundamentals of genetic algorithms: Architecture, Algorithms, and Applications, LaureneFausett, 2002.
Recommended books and references (scientific journals, reports...)	- Practical Genetic Algorithms, Randy L. Haupt, 2004
Electronic References, Websites	http://people.revoledu.com/kardi/tutorial/DecisionTree/how-to-usedecision-tree.htm

Course Description Form

46. Course Name: machine learning
47. Course Code: **COM-412**
48. Semester / Year: First Semester / 2024-2025
49. Description Preparation Date: 1/9/2024
50. Available Attendance Forms: Mandatory attendance
51. Number of Credit Hours (Total) / Number of Units (Total) : 2 hours theoretical + 2 hours practical per week / 3 Units
52. Course administrator's name (mention all, if more than one name)
Name: Prof.Dr. Ziyad Tariq Mustafa Al-Ta'i Email: Ziyad1964tariq@uodiyala.edu.iq
53. Course Objectives
Teaching the student machine learning methods (the calculator) that depend on inference and prediction of future events through statistical methods, artificial networks and their practical (software) applications aimed at the development of technological innovations. ... Teaching and educating students on all the necessary and necessary information related to mach learning, which qualifies them to work and research in all areas of research and prediction meth ...
54. Teaching and Learning Strategies

Strategy	1- Enable students to solve problems related to the intellectual framework of the scientific research method. 2- Enable students to solve problems in a scientific manner and purely scientific bases

55. Course Structure

Evaluation method	Education method	Unit name and/or topic	Required learning outcomes	hours	Week
Oral or written test	Electronic lecture using Microsoft Editor	**Introduction to Machine Learning.**		4	**1**
Oral or written test	Electronic lecture using Microsoft Editor	**Machine learning Models**		4	**2**
Oral or written test	Electronic lecture using Microsoft Editor	**Find S Algorithm. And List Eliminate Algorithm.**		4	**3**
Oral or written test	Electronic lecture using Microsoft Editor	**Candidate Elimination Algorithm.**		4	**4**
Oral or written test	Electronic lecture using Microsoft Editor	**Entropy and Information Gain.**		4	**5**
Oral or written test	Electronic lecture using Microsoft Editor	**Decision Tree Algorithm with Example.**		4	**6**
Oral or written test	Electronic lecture using Microsoft Editor	**Introduction to Neural networks.**		4	**7**
		McCulloch Pitts model.		4	**8**
Oral or written test	Electronic lecture using	**Single Neuron Model.**		4	**9**

	Microsoft Editor				
		Multiple Neurons Model.		4	**10**
Oral or written test	Electronic lecture using Microsoft Editor	**Transfer functions.**		4	**11**
Oral or written test	Electronic lecture using Microsoft Editor	**Single layer Model.**		4	**12**
Oral or written test	Electronic lecture using Microsoft Editor	**Multiple Layers Model.**		4	13
Oral or written test	Electronic lecture using Microsoft Editor	**Learning and Adaptation.**		4	14
Oral or written test	Electronic lecture using Microsoft Editor	**Habbeian Learning Rule. And Perceptron Learning Rule.**		4	15
Oral or written test	Electronic lecture using Microsoft Editor	**Delta Learning Rule. And Widrow Hoff Learning Rule**		4	16

56.Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, dailyoral, monthly, or written exams, reports .... etc

57.Learning and Teaching Resources

Required textbooks (curricular books, if an	Machine Learning, Tom Mitchell, McGraw H Press, 1997

Main references (sources)	**Fundamantals of Neural Networks: Architecture, Algorithms, and application. By Laurene Fausett**, 2010.
Recommended books and references (scientific journals, reports...)	**COS 511: Theoretical Machine Learning**
Electronic References, Websites	**http://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf**

Course Description Form

58. Course Name:	
Mobile Computing	
59. Course Code:	
COM23	
60. Semester / Year:	
2024-2025	
61. Description Preparation Date: 1/9/2024	
62. Available Attendance Forms:	
63. Number of Credit Hours (Total) / Number of Units (Total) **3**	
64. Course administrator's name (mention all, if more than one name)	
Name: *Khalid M.S. Al Zaidi* Email: dr.khaledmoh@uodiyala.edu.iq	
65. Course Objectives	
Course Objectives	• **Students taking this course will develop an understandi of the ways that mobile technologies can be used for teaching and learning. They will also consider the impact of mobile computing on the field of education.** • **To understand concepts of Mobile Communication.(Understand) • To analyse next generation Mobile Communication System. (Analyze) • To understand Mobile networkand transport layers of Mobile**

	Communication. (Understand) of all layers for mobile and ad hoc wireless communication networks. (Analyze) understand IP and TCP layers of Mobile Communication. (Understand)	**• Analyze various protocols • Mobile**

66. Teaching and Learning Strategies

Strategy	

67. Course Structure

Week	**Hours**	**Required Learning Outcomes**	**Unit or subject name**	**Learning method**	**Evaluation method**
1	**3**	Introduction to mobile computing		**Lecture Base**	
2	**3**	Limitations of Mobile Computing		**Lecture Base**	
3	**3**	Mobile Communication		**Lecture Base**	
4	**3**	Mobile Communication **The cellular concept: Cellular system**	Hexagonal geometry cell and concept of frequency reuse, Channel Assignment Strategies Distance to frequency reuse ratio	**Lecture Base**	
5	**3**			**Lecture Base**	
6	**3**	**Telecommunication System:**	GSM: -Channel location, call routing Architecture, PLMN interface, addresses and identifiers, network aspects, frequency allocation, authentication and security, Handoffs Technology. GPRS: network operation, data services, Applications, Billing and charging	**Lecture Base**	
7	**3**	**Mobile IP:**	**Overview of Traditional TCP Need of mobile IP, IP packet delivery, Agent Discovery, Registration, Tunnelling and encapsulation, Route optimization, IP Handoff**	**Lecture Base**	
8	**3**	**Mobile Transport Layer:**	**Overview of Traditional TCP and implications of mobility control. Improvement of TCP: Indirect TCP, Snoop TCP, Mobile TCP, Fast Retransmit/fast recovery, Time-out freezing, Selective retransmission, Transaction-oriented TCP**	**Lecture Base**	
9	**3**	**Wireless Application Protocol:**	**Introduction of WAP, WAP applications, WAP Architecture, WAP Protocol Stack, Challenges in WAP**	**Lecture Base**	
10	**3**	**Mobile Ad Hoc wireless networks:**	**Introduction, Benefits, Difference, Routing protocols for ad hoc wireless networks: DSDV and AODV**	**Lecture Base**	
11	**3**	**Introduction to 4G:**	**Introduction, features and challenges, Applications of**	**Lecture Base**	

			4G, 4G network architecture		
12	**3**	**Mobile Device Operating Systems**		**Lecture Base**	
13	**3**	**Mobile Operating Systems**	**Software Development Kit: iOS, Android, BlackBerry, Windows Phon**	**Lecture Base**	
14	**3**	**Mobile Payment System — Security Issues**		**Lecture Base**	
15	**2**			**Lecture Base**	

68.Course Evaluation

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc. b. The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room. c. Practical examination will be conducted at the end of the semester for evaluation of the performance of students in the laboratory. d. Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory

69.Learning and Teaching Resources

Required textbooks (curricular books, if any)	Tomasz Imielinski and Henry F. Korth,"MOBILE COMPUTING".
Main references (sources)	- *Ivan Stojmenovic´,"Handbook of Wireless Networks and Mobile Computing".* - *Martyn Mallick; "Mobile and Wireless Design Essentials ".* • *Behrouz A. Forouzan; "Data Communication and networking".*
Electronic References, Websites	- https://www.udemy.com http://www.protocols.com/ 3. https://developer.apple.com/ 4. https://www.udemy.com 5. http://nptel.ac.in

Course Description Form

70.Course Name:

Modeling and Simulation

71.Course Code:

COM-424

72.Semester / Year: 2024/2025

73.Description Preparation Date: 1/9/2024

74.Available Attendance Forms:

75. Number of Credit Hours (Total) / Number of Units (Total) **3**

76. Course administrator's name (mention all, if more than one name)

Name: ***Dhahir Abdulhadi Abdullah***
Email: **dhahair@uodiyala.edu.iq**

77. Course Objectives

Course Objectives	• This course provides an introduction to the process of designing model existing or proposed real-world systems, and how to use the models perform simulations that allow for predictions about the future behavio the system. The system could be something as mundane as a cricket ma to something more complex, such as a communication network, transportation system. Most systems of interest will require development of one or more statistical models. Thus, modeling simulation has a significant overlap with probability and statistics. course topics will include a review of concepts from probability statistics that are relevant to modeling and simulation, algorithms random-variable sampling, modeling and analysis of basic queue systems, variance-reduction techniques, statistical-validation techniqu Independent Monte Carlo (IMC) and Markov-Chain Monte Carlo (MC simulations, and discrete-event modeling and simulation. Programm assignments will be provided throughout the semester. In addition, e student will complete an end-of-term project that centers on the mode and simulation of a system of interest.

78. Teaching and Learning Strategies

Strategy	The goal of this course is to enable the student to design, develop, implement, and anal simulation models, and to have comprehensive of important aspects of simulation study includi modeling, simulation software, model verification and validation, random number generation variates, and analysis of simulation experiment and applications.

79. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1	**3**	**Flowchart for departure event routine**		**Lecture Base**	
2	**3**	**SIMULATION PROGRAMMING**		**Lecture Base**	
3	**3**	**Arrival and departure event function**		**Lecture Base**	
4	**3**	**Arrival and departure event function** Memory protection		**Lecture Base**	
5	**3**			**Lecture Base**	
6	**3**			**Lecture Base**	

		Arrival and departure event function			
7	**3**	**Exponential, Poisson, normal, uniform, and Binomial distribution.**		**Lecture Base**	
8	**3**	**RANDOM NUMBER GENERATION**		**Lecture Base**	
9	**3**	**Chi-square test and K-S test basis**		**Lecture Base**	
10	**3**	• **Markov-Chain Monte Carlo methods**		**Lecture Base**	
11	**3**	• **Simulated annealing**		**Lecture Base**	
12	**3**	• **Statistical analysis of simulated data**		**Lecture Base**	
13	**3**	• **Statistical validation techniques**		**Lecture Base**	
14	**3**	**Queuing Theory**		**Lecture Base**	
15	**2**			**Lecture Base**	

80.Course Evaluation

The course is an introduction to modeling and simulation. It includes the following topics: Fundamental concepts of computer simulation; Models for computer simulation; Random numbers: pseudorandom number generation and testing, Monte Carlo methods; Introduction to distribution functions; Simulation modeling; Discrete-event simulation; Continuous simulation; Verification and validation of simulation models; Input analysis; Output analysis; Queuing theory models; Design code; Test and Debug simulation programs; Sample of applications. There will be weekly practice in the lab.

81.Learning and Teaching Resources

Required textbooks (curricular boo if any)	Simulation Modeling and Analysis , 5/e, by Averil M Law and W. David Kelton, McGraw Hill, 2015. www.mhhe.com/engcs/industrial/lawkelton 2- Tayfur Altiok and Benjamin Melamed, Simulation Modeling and Analysis with Arena, edition 2007/ or the latest.
Main references (sources)	• Simulation Modeling and Analysis , 5/e, by Averil M Law and W. David Kelton, McGraw Hill, 2015. www.mhhe.com/engcs/industrial/lawkelton 2- Tayfur Altiok and Benjamin Melamed, Simulation Modeling and Analysis with Arena, edition 2007/ or the latest.
Electronic References, Websites	**http://www2.mansfield.edu/mathematics/program-course-goals-objectives-and-outcomes/index.cfm**

Course Description Form

82.Course Name:
Multimedia
83.Course Code:
COM-414
84.Semester / Year:
first/2024-2025
85.Description Preparation Date:
1/9/2024
86.Available Attendance Forms:
Presence
87.Number of Credit Hours (Total) / Number of Units (Total)
30/3
88. Course administrator's name (mention all, if more than one name)
Name: Muna Rashid Hameed Email: munarashid@uodiyala.edu.iq
89.Course Objectives

Course Objectives	• This course aims to introduce the fundamental elements of multimedia. It will provide an understanding of the fundamental elements in multimedia. The emphasis will be on learning the representations, perceptions and applications of multimedia. Software skills and hands on work on digital media will also be emphasized. On completion of the subject, the students will understand the technologies behind multimedia applications and master the skills for developing multimedia projects. After successfully completing the module student should be able to: • Summarize the key concepts in current multimedia technology. •

	Create quality multimedia software titles.

90.Teaching and Learning Strategies

Strategy	• students will have developed a comprehensive understanding of the principles and techniques involved in creating and analyzing multimedia content. They will be equipped with the knowledge and skills to effectively utilize multimedia tools and technologies to communicate ideas, information, and stories across various digital platforms. • Students will demonstrate proficiency in designing, producing, and evaluating multimedia projects, incorporating elements such as text, images, audio, video, and interactive elements. Additionally, they will have cultivated critical thinking, problem-solving, and creativity in the context of multimedia production, enabling them to adapt to evolving technologies and effectively engage diverse audiences in today's digital landscape.

91. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1	3		An Introduction to MULTIMEDIA	Electronic lec using Micros Editor	Oral or written test
2	3		Multimedia Software Tools	Electronic lect using Micros Editor	Oral or written test
3	3		Overview of Multimedia Software Tools	Electronic lect using Micros Editor	Oral or written test
4	3		Multimedia Production	Electronic lect using Micros Editor	Oral or written test
5	3		Multimedia Production	Electronic lect using Micros Editor	Oral or written test
6	3		Graphics/Image Data Types	Electronic lect using Micros Editor	Oral or written test
7	3		Gamma Correction	Electronic lect using Micros Editor	Oral or written test
8	3		Color Models in Video	Electronic lect using Micros Editor	Oral or written test
9	3		Chroma Subsampling	Electronic lect using Micros Editor	Oral or written test

10	3		Audio Digitization	Electronic lect using Micros Editor	Oral or written test
11	3		Audio Filtering	Electronic lect using Micros Editor	Oral or written test
12	3		Quantization and Transmission of Audio	Electronic lect using Micros Editor	Oral or written test
13	3		Color Histogram	Electronic lect using Micros Editor	Oral or written test
14	3		Texture Layout	Electronic lect using Micros Editor	Oral or written test
15	3		Image Retrieving	Electronic lect using Micros Editor	Oral or written test

92.Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, daily oral, monthly, or written exams, reports .... etc

93.Learning and Teaching Resources

Required textbooks (curricular books, if any)	
Main references (sources)	- "Fundamentals of Multimedia", by Ze-N Li, Mark S Drew, Published by Prentice Hall 2004
Recommended books and references (scientific journals, reports...)	
Electronic References, Websites	1. W. K. Pratt ,Digital Image Processing, Second Ed. Wiley, (1991). 2. Gonzalez "Digital image processing" ,2008. D.Philips, "image processing in c language", second edition, April 2000.

Course Description Form

94. Course Name: Network Security

95. Course Code: **COM-423**

96. Semester / Year: Semester 2024/20285

97. Description Preparation Date: 1/9/2024

98. Available Attendance Forms: Mandatory attendance

99. Number of Credit Hours (Total) / Number of Units (Total) : 3 hours theoretical + p**e**r week / 3 Units

100. Course administrator's name (mention all, if more than one name)

Name: Dr. Burhan Molan saleh
Email: **burhan@uodiyala.edu.iq**

101. Course Objectives

Network Security Fundamentals is a basic course in network security. The student will learn the key elements of network security scientifically and internet security specifically, such as potential attack types, as well as types and methods of web protection, various algorithms, and protocols used in network security, and what session security entails. The curriculum aims to:

1- The desired goal for the student to successfully pass the course requirements is for the student to understand the principles of network security and the importance of using them over the internet, in addition to the applied fields in which this field is involved.

2- The student's understanding of the types and methods of research.

3- Developing the student's ability to use available software in this field in addition to the skills acquired in programming and encryption.

102. Teaching and Learning Strategies

Strategy	1- Enable students to solve problems related to the intellectual framewor**k** of the scientific research method.

	2- Enable students to solve problems in a scientific manner and on pu scientific bases

103. Course Structure

Evaluation method	Education method	Unit name and/or topic	Required learning outcomes	hours	Week
Oral or written test	Electronic lecture using Microsoft Editor	**the basics of the Internet security**		4	**1**
Oral or written test	Electronic lecture using Microsoft Editor	**exploring the attacks techniques that can be used by attackers**		4	**2**
Oral or written test	Electronic lecture using Microsoft Editor	**the ways of defense.**		4	**3**
Oral or written test	Electronic lecture using Microsoft Editor	**Control hijacking attacks**		4	**4**
Oral or written test	Electronic lecture using Microsoft Editor	**Exploitation techniques and fuzzying;**		4	**5**
Oral or written test	Electronic lecture using Microsoft Editor	**Secure system design**		4	**6**
Oral or	Electronic lecture	**access control.**		4	**7**
		protection Tools		4	**8**

		for writing robust application code			
Oral or	Electronic lecture	**web security models**		4	**9**
		User authentication		4	**10**
Oral or written test	Electronic lecture using Microsoft Editor	**session management**		4	**11**
Oral or written test	Electronic lecture using Microsoft Editor	**Cross-Site Attacks; SQL Injection Attacks;**		4	**12**
Oral or written test	Electronic lecture using Microsoft Editor	**What is SSL;**		4	13
Oral or written test	Electronic lecture using Microsoft Editor	**What is HTTPS, how it works, its pitfalls;.**		4	14
Oral or written test	Electronic lecture using Microsoft Editor	**What is certificates, where it can be used and how can created;**		4	15
Oral or written test	Electronic lecture using Microsoft Editor	**Delta Learning Rule. And Widrow Hoff Learning Rule**		4	16

104. Course Evaluation

Practical exams
2- Theoretical exams
3- Reports and studies
4- Daily quizzes with self-solvable questions
5- Grades determined by homework assignments

105. Learning and Teaching Resources

Cryptography and network security, 5th 1 - Edition, William Stallings, 2011	Machine Learning, Tom Mitchell, McGraw Press, 1997
Network Security and Management. PHI Learning Pvt. Ltd., 2011.	**Fundamantals of Neural Networks: Architecture, Algorithms, and applica tion By Laurene Fausett**, 2010.
***Guide to computer network security*. Heidelberg, Germany: Springer, 2013.**	**COS 511: Theoretical Machine Learni g**
https://www.cisco.com/c/ar_ae/products/security/what-is-cybersecurity.html	**http://www.cs.princeton.edu/courses/arch ve/spr08/cos511/scribe_notes/0204.pdf**

Course Description Form

1. Course Name: Operating System1

2. Course Code: COM-411

3. Semester / Year: First Semester / 2024-2025

4. Description Preparation Date: 1/9/2024

5. Available Attendance Forms: Mandatory attendance

6. Number of Credit Hours (Total) / Number of Units (Total): 2 theoretical hours - 2 practical hours / a week - 3 units

7. Course administrator's name (mention all, if more than one name)

Name: Prof.Dr. Jamal Mustafa Abbas Al-Tuwaijari
Email: dr.altuwaijari@uodiyala.edu.iq

8. Course Objectives

Course Objectives This academic curriculum is prepared introduce the student to operating syster their definition, structure, development, a tasks. The course addresses the structure operating systems and the role of operati systems, operating system structur processes and their synchronization, ta scheduling, main memory and virtu memory management. The curriculum ai to: -	**1. Prepare the student to understand the basic conce ots operating systems and their applications.** **2. Providing students with basic knowledge of the structure of operating systems, which includes the hardware and software components of the computer** **3. Improving the student's level and providing her/ him with the scientific skills and abilities to understand methods for managing the main memory of the com ute as well as the case for virtual memory and temporar y memory.** **4. Developing the student's ability to use software a d applications available in the field of computers that re compatible with the operating systems used in the computer, in addition to the skills in using central processing unit management software.** **5. Enabling the student to understand infor mat management software (file system) as well as dea l w problems related to processors and operations**

9. Teaching and Learning Strategies

Strategy	1.Building students' basic knowledge about operating systems, th ei principles, basic concepts, and types. 2. Enabling students to acquire mental, cognitive, analytical and understanding skills in the field of operating systems. 3. Enabling students to solve problems related to the intelle ctu framework of operating systems management

10. Course Structure	

Evaluation method	Education method	Unit name and/or topic	Required learning outcomes	hours	Week
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Introduction ,Role and purpose of the operating system	Definitions and a general introduction to operating systems, during which the notes and instructions that students must adhere to in order to achieve the best performance in studying the subject are explained.	2	**1**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	History and types of operating system, Operating system development	History and types of operating system	2	**2**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Functionality of a typical operating system	Functionality of operating system	2	**3**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Computer System, Operating System Components and structure	Operating System Components	2	**4**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Performance and Development of operating system	Development of operating system	2	**5**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	operating system functions and services	operating system functions	2	**6**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Processes, process concepts, operation on processes	Processes	2	**7**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Interrupts: methods and implementations	Interrupts	3	**8**
Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Concept of user/system state and protection, transition to kernel mode	Concept of user/system	2	**9**

	Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Concurrency Context switching Program status words (PSWs)	Concurrency Context switching Program status words (PSWs)	2	**10**
	Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Threads, Thread structure	Threads and Thread structure	2	**11**
	Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Scheduling and policies, Scheduling levels	Scheduling and Scheduling levels	2	**12**
	Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Dispatcher Scheduling criteria	Dispatcher Scheduling criteria	2	**13**
	Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Preemptive and non-preemptiv scheduling	Preemptive and non-preemptiv scheduling	2	**14**
	Daily exams and homework, in addition to monthly exams	Electronic lecture using Microsoft Editor	Scheduling Algorithms	Scheduling algorithms	2	**15**

11.Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, dailyoral, monthly, or written exams, reports .... etc, in addition to practical exams.

12.Learning and Teaching Resources

Required textbooks (curricular books, if an	Not available
Main references (sources)	Operating System Concepts. By SilBerschtz and Galvin
Recommended books and references (scientific journals, reports...)	Operating Systems, by Harvey M. Deitel, Paul J. Deitel, and David R. Choffnes
Electronic References, Websites	http://www.deitel.com/Books/OperatingSystems/

Course Description Form

106. Course Name:

Data mining

107. Course Code: COM-423

108. Semester / Year: 2024/2025

2024-2025

109. Description Preparation Date:

110. Available Attendance Forms:

111. Number of Credit Hours (Total) / Number of Units (Total) **2**

112. Course administrator's name (mention all, if more than one name)

Asst.Prof.Dr. Muntadher khamees
Email: alkarawis@uodiyala.edu.iq

113. Course Objectives

Course Objectives	This course introduces basic concepts, tasks, methods, and techniques in Data Mini The emphasis is on various Data Mining problems and their solutions. Students develop an understanding of the Data Mining and issues, learn various techniques Data Mining, and apply the techniques in solving Data Mining problems using tools systems. Students will also be exposed to a sample of Data Mining applications.

114. Teaching and Learning Strategies

Strategy	data is stored, analyzed, and disseminated via data mining , a interdisciplinary field consisting both data mining and computer science. An application of data minig is to determine the funct of genes and proteins, to establish evolutionary relationships, and to calculate the high dimensio shape of proteins by using computer progra

115. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1	**3**	Introduction	Chap.1 Ref. 1	**Lecture Base**	
2	**3**	Data	Chap.2 Ref. 1, Chap.2, Ref 5.	**Lecture Base**	
3	**3**	Data Preprocessing I	Ch.7, Ref. 5, Chap.2 Ref. 1, Chap.2 Ref. 6	**Lecture Base**	
4	**3**	Data Preprocessing II Data Visualization	Ch.7, Ref. 5, Chap.2 Ref. 1, Chap.2 Ref. 6	**Lecture Base**	
5	**3**			**Lecture Base**	
6	**3**	Classification I: Basic Concepts, Decision Trees, and Model Evaluation	Chap.3 Ref.1 Chap.4 Ref.1	**Lecture Base**	
7	**3**	Classification II : Alternative Techniques	Chap.4,5 Ref.1	**Lecture Base**	

8	**3**	Clustering: Basic concepts	Chap.8 Ref.1	**Lecture Base**	
9	**3**	Clustering Algorithms I: Sequential algorithms, Hierarchical algorithms	Chap.8,9 Ref.1	**Lecture Base**	
10	**3**	Regression Analysis		**Lecture Base**	
11	**3**	Mining Frequent Patterns, Associations, and Correlations I	Chap.6 Ref.1	**Lecture Base**	
12	**3**	Mining Frequent Patterns, Associations, and Correlations II	Chap.6 Ref.1	**Lecture Base**	
13	**3**	Advance Topics		**Lecture Base**	
14	**3**	Review		**Lecture Base**	
15	**2**	First Exam	Exam 2	**Lecture Base**	

116. Course Evaluation

The course serves as an introduction to data mining and applications . The aim is that students should understand how data mining can be applied and evaluated and provide tools for practical approaches to mining algorithm

117. Learning and Teaching Resources

Required textbooks (curricular books, any)	. Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques" Third Edition, Elsevier, 2012.
Main references (sources)	1- . Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to data mining," 2006. 2- Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques," Second Edition, Elsevier Inc., 2006. 3- Anil K. Jain, Richard C. Dubes, "Algorithms for Clustering Data," Prentice-Hall Inc., 1988. 4- David Hand, Heikki Mannila, Padhraic Smyth, "Principles of Data Mining,"The MIT Press, 2001. 5- Ian H. Witten, Eibe Frank, "Data Mining, Practical Machine Learning Tools and Techniques," Morgan Kaufmann Publishers, 2005. 6- Daniel T. Larose, "Discovery Knowledge in Data, An Introduction to Data Mining," A John Wiley & Sons, Inc., Publication, 2005. 7- Further Readings are also preferable. •
Electronic References, Websites	1.https://www.coursera.org/lecture/code-free-data-science/introduction-to-data-mining-hbb2V 2. https://onlinecourses.swayam2.ac.in/cec19_cs01/preview material

Course Description Form

118.	Course Name: **Web applications Development**
119.	Course Code: **COM-322**
120.	Semester / Year: / 2024-2025
121.	Description Preparation Date: 1/9/2024
122.	Available Attendance Forms: Mandatory attendance
123.	Number of Credit Hours (Total) / 2 hours theoretical per week / 3 Units
124.	Course administrator's name (mention all, if more than one name)
	Name: Assist. Prof. Dr. Dheyab Salman Ibrahim Al-nedawy Email: dr.dheyab@uodiyala.edu.iq
125.	Course Objectives
	Teaching the student web design technologies such as HTML, CSS, and JavaScript in orde to create different HTML documents and teaching the students to build website. Teaching and educating students on all the necessary and necessary information related to w design techniques, which qualifies them to work and research in all areas of research.
126.	Teaching and Learning Strategies
Strategy	1- Enable students to create web pages based on HTML tags. 2- Teach students to format these web pages based on CSS rules. 3- Teach students to increase the inactive these web pages based on JavaScrip language.

10. Course Structure

Evaluation method	Education method	Unit name and/or topic	Required learning outcomes	hours	Week
Oral or written test	Electronic lecture using Microsoft Editor	Introduction to a Computer Networks		2	**1**
Oral or written test	Electronic lecture using Microsoft Editor	Internet and Web: Overview		2	**2**
Oral or written test	Electronic lecture using Microsoft Editor	Web Design Technologies		2	**3**
Oral or written test	Electronic lecture using Microsoft Editor	HTML Tags		2	**4**
Oral or written test	Electronic lecture using Microsoft Editor	HTML Links		2	**5**
Oral or written test	Electronic lecture using Microsoft Editor	HTML Lists		2	**6**
Oral or written test	Electronic lecture using Microsoft Editor	HTML Tables		2	**7**
Oral or written test	Electronic lecture using Microsoft Editor	HTML Forms		2	**8**
Oral or written test	Electronic lecture using Microsoft Editor	CSS Rules		2	**9**
Oral or written test	Electronic lecture using Microsoft Editor	CSS – Selectors		2	**10**
Oral or written test	Electronic lecture using Microsoft Editor	CSS – Inclusion: Inline, Internal ways		2	**11**
Oral or written test	Electronic lecture using Microsoft Editor	CSS – Inclusion: External way		2	**12**

11. Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, daily oral, monthly, or written exams, reports .... etc.

12. Learning and Teaching Resources

Required textbo (curricular book any)	How the Internet Works, Preston Gralla, Pearson Education, Eighth Edition.
Main referen (sources)	Internet for Everyone, Alexis Leon, S. Chand (G/L) & Company Ltd; Second - .Edition
Recommended books and references (scientific journals, reports...)	Web Design with HTML and CSS Digital Classroom. https://bawar.net/data0/books/5d626a582939a/pdf/web-design-with-html-an l-css-digital-classroom.pdf
Electronic References, Websites	Learning Web Design Fourth Edition A Beginner's Guide to HTML, CSS, JavaScript and Web Graphics. https://wtf.tw/ref/robbins.pdf

Course Description Form

1. Course Name:

Research methodology

2. Course Code:

COM-316

3. Semester / Year:2024/2025

Second semester/third year

4. Description Preparation Date:1/9/2024

5. Available Attendance Forms:

6. Number of Credit Hours (Total) / Number of Units (Total)

2 hours/1units

7. Course administrator's name (mention all, if more than one name)

Name: layla abd al.haq esmaeel
Email: laylaaeabdalhaq@uodiyala.edu.iq

8. Course Objectives

Course Objectives	1. The objective required of the student in order to successfully pass the requirements of the course is the student's awareness of the importance of scientific research and scientific methods. used in carrying out research and auxiliary programs. 2.The student's awareness of the types of scientific research and each specialization has a method in scientific research. 3. Develop the student's ability in scientific research and teach him the basics of scientific research and the ethics of scientific research

9. Teaching and Learning Strategies

Strategy	- Using various modern presentation tools - the lecture method and the use of the interactive whiteboard - providing students with the basics and additional topics related to scientific research - Forming discussion groups during the lectures to discuss modern systems that require - thinking and analysis Asking the students a set of thinking questions during the lectures such as what, how, when and why For specific topics - giving students homework that requires subjective explanations in cau ways

10. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method

1	3	Definitions and a general introduction Explanation of the necessary notes and instructions The students and the professor are obligated To achieve the best performance from studying the subject		Electronic Lecture using editor Microsoft	Oral or writ test
2	3	Meaning Of Research		Electronic Lecture using editor Microsoft	Oral or writ test
3	3	General CharacteristicOf Research		Electronic Lecture using editor Microsoft	Oral or writ test
4	3	Types Of Research		Electronic Lecture using editor Microsoft	Oral or writ test
5	3	Research Problem		Electronic Lecture using editor Microsoft	Oral or writ test
6	3	Problem Formulation		Electronic Lecture using editor Microsoft	Oral or writ test
7	3	High Impact ResearchTools		Electronic Lecture using editor Microsoft	Oral or writ test
8	3	Project Planning		Electronic Lecture using editor Microsoft	Oral or writ test
9	3	Gantt Chart		Electronic Lecture using editor Microsoft	Oral or writ test
10	3	Research *Ethics*		Electronic Lecture using editor Microsoft	Oral or writ test
11	3	Writing the LiteratureReview		Electronic Lecture using editor Microsoft	Oral or writ test
12	3	Citation ManagementTools		Electronic Lecture using editor Microsoft	Oral or written test
13	3	Methods of DataCollection		Electronic Lecture using editor Microsoft	Oral or written test
14	3	Research Report		Electronic Lecture using editor Microsoft	Oral or written test

15	3	General Format of Rese Report		Electronic Lecture using editor Microsoft	Oral or written test

11.Course Evaluation
- Practical tests 2- Theoretical tests 3- Reports and studies 4- Daily exams with self- solving questions 5- Marks specific to homework

12.Learning and Teaching Resources	
Required textbooks (curricular books, if any)	* Research Methodology a step-by-step guide for beginners, Ranjit Kumar, 3rd edition, 2011.
Main references (sources)	
Recommended books and references (scientific journals, reports...)	
Electronic References, Websites	1- Research Methodology - Methods and Techniques, C.R. Kothari, 3Sd edition, 2004. 2- Fundamental of Research Methodology and Statistics, Yogesh Kumar Singh, 2006. 3- How to do the Final Year Projects A practical Guideline for Computer Science and ITStudents, Hossein Hassani, 2012.

Course Description Form

127.	Course Name:
	COM-311 Language Translator
128.	Course Code:
	COM-311
129.	Semester / Year:
	2024-2025
130.	Description Preparation Date:
	1/9/2024
131.	Available Attendance Forms:
	Presence
132.	Number of Credit Hours (Total) / Number of Units (Total)
	60

133. Course administrator's name (mention all, if more than one name)

Name: **Wasan ahmed ali**
Email: **wasanahmed83@gmail.com**

134. Course Objectives

Course Objectives	Introducing the student to the six basic stages of the compiler and how each stage works to understand how to convert basic code into a computer program. A- **Cognitive objectives** 1- Definition of the program code 2- Using the code to serve the system 3- Defining the important steps in designing the code 4- Identify the types of errors that may occur during design **B- The skills objectives of the course** 1- Knowledge skills - remembering 2- Memorization and analysis skills 3- Use and development skills

135. Teaching and Learning Strategies

Strategy	The main strategy that will be adopted in delivering the Language Translator module is to engage students actively in practical exercises to enhance their understanding and develop their critical thinking skills. To enhance learning and encourage active participation, interactive lessons will be conducted. These tutorials will include practical exercises where students will work on applying all the methods used in the compiler and all algorithms of the parser as programs in one of the developed programming languages. This hands-on approach will help students understand the practical implications of theoretical concepts discussed in the classroom. Overall, the unit delivery approach aims to actively engage students, improve their critical thinking skills, and provide them with hands-on experiences in image processing and a language interpreter. By combining theoretical knowledge with practical activities, students will develop a deeper understanding of language compiler concepts and their applications in various fields.

136. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method

1	2		Shift-Reduce Parsing	In-person lecture using computer, display screen, pen and blackboard	Oral or written tes
2	2		Handles	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
3	2		Stack Implementation of Shift-Reduce Parsing	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
4	2		Operator-Precedence Pars (OPP) Table Construction of Operator-Precedence Relations	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
5	2			In-person lecture u computer,display screen, pen blackboard	Oral or written tes
6	2		Relations of Operator-Precedence Table	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
7	2		LR parser	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
8	2		SLR Parser	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
9	2		Canonical LR Parser LALR Parser	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
10	2		Conflict types	In-person lecture u computer,display screen, pen blackboard	Oral or written tes
11	2			In-person lecture u computer,display screen, pen blackboard	Oral or written tes
12	2		Shift - Reduce Conflict Reduce - Reduce Conflict	In-person lecture using compu display screen, pen blackboard	Oral or written tes
13	2			In-person lecture using compu display screen, pen blackboard	Oral or written tes
14	2		Semantic Errors		Oral or written tes
15	2				Oral or written tes

137. Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, daily oral, monthly, or written exams, reports .... etc

138. Learning and Teaching Resources

Required textbooks (curricular books, if any)	
Main references (sources)	Compiler principles and tools
Recommended books and references (scientific journals, reports...)	
Electronic References, Websites	

Course Description Form

139. Course Name:	
Artificial Intelligence	
140. Course Code: COM-312	
141. Semester / Year:	
2024-2024	
142. Description Preparation Date: **1/9/2024**	
143. Available Attendance Forms:	
144. Number of Credit Hours (Total) / Number of Units (Total)	
145. Course administrator's name (mention all, if more than one name)	
Name: Dr.Adil Abdulwahhab Al-Azzawi Email: adil_alazzawi@updiyala.edu.iq	
146. Course Objectives	
Course Objectives	• **The course begins by describing what the latest generation of artifi intelligence techniques can do. After an introduction to some basic conce and techniques, the course illustrates both the potential and curr limitations of these techniques with examples from a variety applications. We spend some time on understanding the strengths weaknesses of human decision-making and learning, specifically combination with AI systems. Exercises will include hands-on applicatio basic AI techniques as well as selection of appropriate technologies for a gi problem and anticipation of design implications. In a final project, group students will participate in the creation of an AI-based application.**
147. Teaching and Learning Strategies	
Strategy	**The course begins by describing what the latest generation of artificial intellige techniques can do. After an introduction to some basic concepts and techniques, course illustrates both the potential and current limitations of these techniques v examples from a variety of applications. We spend some time on understanding strengths and weaknesses of human decision-making and learning, specifically combination with AI systems and on ethical and policy implications of new AI capabilit Exercises will include hands-on application of basic AI techniques as well as selectio appropriate technologies for a given problem and anticipation of design implications a final project, groups of students will participate in the creation of an AI-ba application.**

148. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method

1	**2**	**Knowledge Representation**	**Introduction to artificial intelligence**	**Lecture Base**	
2	**2**		**Problem solving in A.I**	**Lecture Base**	
3	**2**		**Problem solving in A.I (cont.)**	**Lecture Base**	
4	**2**		**Importance of search for AI**	**Lecture Base**	
5	**2**		**Uninformed and informed search**	**Lecture Base**	
6	**2**		**Adversarial search**	**Lecture Base**	
7	**2**		**Mid-term Exam**	**Lecture Base**	
8	**2**	**Apply selected basic AI techniques, judge applicability of more advanced techniques.**	**Local search (gradient descent)**	**Lecture Base**	
9	**2**		**Reasoning with Uncertainty**	**Lecture Base**	
10	**2**		**Bayesian networks**	**Lecture Base**	
11	**2**	**Participate in the design of systems that act intelligently and learn from experience.**	**Robotic control and planning**	**Lecture Base**	
12	**2**		**Robotic control and planning (cont.)**	**Lecture Base**	
13	**2**		**Knowledge Representation**	**Lecture Base**	
14	**2**		**Knowledge Representation**	**Lecture Base**	
15	**2**		**Final Exam Review**	**Lecture Base**	

149. Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, daily oral, monthly, or written exams, reports .... etc

150. Learning and Teaching Resources

Required textbooks (curricular books, if any)	Russell & Norvig, Chapter 1, "Introduction" Artificial Intelligence: A Modern Approach, 202
Main references (sources)	Adversarial Search" in Russell & Norvig, Artifi Intelligence: A Modern Approach, 2020
Recommended books and references (scientific journals, reports...)	
Electronic References, Websites	

Course Description Form

151. Course Name:

Pattern Recognition

152. Course Code:

COM-326

153. Semester / Year:

2024-2025

154. Description Preparation Date: **1/9/2024**

155. Available Attendance Forms:

156. Number of Credit Hours (Total) / Number of Units (Total)

157. Course administrator's name (mention all, if more than one name)

Name: Dr.Adil Abdulwahhab Al-Azzawi
Email: adil_alazzawi@updiyala.edu.iq

158. Course Objectives

Course Objectives	• **Understand the basic concepts and principles of pattern recogniti including feature extraction, classification, and clustering.** • **Apply various pattern recognition algorithms and techniques, includ supervised and unsupervised learning methods, to solve real-world proble** • **Analyze and evaluate the performance of pattern recognition systems us appropriate metrics and evaluation methods.** • **Gain practical experience in implementing pattern recognition algorith using programming languages such as Python or MATLAB.** • **Explore advanced topics in pattern recognition, such as deep learni probabilistic graphical models, and ensemble methods.** • **Apply pattern recognition techniques to different domains, including im processing, natural language processing, and bioinformatics.** • **Develop critical thinking and problem-solving skills through hands projects, assignments, and case studies.** • **Understand the ethical and societal implications of pattern recognit technologies, including privacy concerns and bias in decision-mak systems.**

159. Teaching and Learning Strategies

Strategy	**Pattern recognition theory and practice is concerned with the design, analysis, a development of methods for the classification or description of patterns, objects, sign and processes. At the heart of this discipline is our ability to infer the statistical behav of data from limited data sets, and to assign data to classes based on generalized noti of distances in a probabilistic space. Many commercial applications of pattern recognit exist today, including voice recognition (e.g., Amazon Alexa), fingerprint classificat (e.g., MacBook Pro touch bar), and retinal scanners (e.g., your favorite cheesy s movie).**

160. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1	**2**	**Understands basic structure of pattern recognition systems.**	**Introduction: An Overview of Machine Learning**	**Lecture Base**	
2	**2**		**Decision Theory: Bayes Rule**	**Lecture Base**	
3	**2**		**Decision Theory: Gaussian Classifiers**	**Lecture Base**	

4	**2**	**Expresses principal units within a pattern recognition system.**	**Decision Theory: Generalized Gaussian Classifiers**	**Lecture Base**	
5	**2**	**Summarizes execution of a pattern recognition system.**	**Parameter Estimation: The Bayesian Approach**	**Lecture Base**	
6	**2**		**Mid-term Exam**	**Lecture Base**	
7	**2**		**Decision Theory: Discriminant Analysis**	**Lecture Base**	
8	**2**		**Parameter Estimation: The Expectation Maximization Theorem**	**Lecture Base**	
9	**2**	**Defines the relationship between pattern and feature.**	**Parameter Estimation: Discriminative Training**	**Lecture Base**	
10	**2**		**Experimental Design: Foundations of Machine Learning**	**Lecture Base**	
11	**2**	**Explains supervised and unsupervised pattern recognition approaches.**	**Experimental Design: Evaluation**	**Lecture Base**	
12	**2**		**Statistical Significance**	**Lecture Base**	
13	**2**	**Analyzes the success of a feature recognition system.**	**Jackknifing, Bootstrapping and Combining Classifiers**	**Lecture Base**	
14	**2**		**Introduction to Nonparametric Techniques**	**Lecture Base**	
15	**2**		**Final Exam Review**	**Lecture Base**	

161. Course Evaluation

Distributing the score out of 100 according to the tasks assigned to the student such as daily preparation, daily oral, monthly, or written exams, reports .... etc

162. Learning and Teaching Resources

Required textbooks (curricular books, if any)	A. Lindholm, N. Wahlstrom, F. Lindsten and T. Schon, Machine Learning: A First Course for Engineering and Scientists, Cambridge University Press, New York, New York, USA, ISBN: 978-1-108-84360-7, pp. 338, 2022. URL: http://smlbook.org/book/sml-book-draft-latest.pdf

Main references (sources)	C.M. Bishop, Pattern Recognition and Machine Learning, Springer, ISBN: 978-0387310732, 2003.
Recommended books and references (scientific journals, reports...)	
Electronic References, Websites	

Bologna Process

Level+A7:T46	Semester	No.	Module Code	Module Name in English	CL (hr/w)	hr/sem		
1	One	1	COS-101	Differentiation Methods	3	125	5.00	B
		2	COM-111	Introduction to Programming	3	200	8.00	C
		3	COM-112	Computational mathematics	3	125	5.00	C
		4	COM-113	Computer Skills	2	100	4.00	C
		5	COM-114	Digital Logic	3	150	6.00	C
		6	UD11	English Language	2	50	2.00	B
					16	750	30.00	
	Semester	No.	Module Code	Module Name in English	Exam hr/sem CL (hr/w)	hr/sem	ECTS	Module Type
	Two	1	COS-102	Integration Methods	3	150	6.00	B
		2	COM-121	Programming Fundamentals	3	200	8.00	C
		3	COM-122	Discrete Structures	3	150	6.00	C
		4	COM-123	Computer Organization	2	150	6.00	C
		5	UD14	Human Rights and democracy	2	50	2.00	B
		6	UD12	Arabic Language	2	50	2.00	B
					15	750	30.00	
Level	Semester	No.	Module Code	Module Name in English	Exam hr/sem CL (hr/w)	hr/sem		
2	Three	1	COM-211	Introduction to Object Oriented Language	3	175	7.00	C
		2	COM-212	Data Structurs	3	150	6.00	C
		3	COM-213	Computation Theory	3	100	4.00	C
		4	COM-214	Web Design and Programming	3	150	6.00	C
		5	SCI-103	Numerical Methods	3	125	5.00	B
		6	UD24	Baath Party Crimes in Iraq	2	50	2.00	B
					17	750	30.00	
	Semester	No.	Module Code	Module Name in English	Exam hr/sem CL (hr/w)	hr/sem		
	Four	1	COM-221	Algorithms Design and Analysis	2	125	5.00	C
		2	COM-222	Computer Graphics	2	150	6.00	C
		3	COM-223	Computer Architecture	2	100	4.00	C
		4	COM-224	Introduction to Python	2	150	6.00	C
		5	COM-225	Object Oriented Programming	2	125	5.00	B

	6	COM-226	**Visual Programming**	2	100	4.00	C
				12	750	30.00	

MODULE DESCRIPTION FORM

Module Information					
Module Title	**Computation Theory**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-213**				
ECTS Credits	**4**				
SWL (hr/sem)	**100**				
Module Level	2		**Semester of Delivery**	3	
Administering Department	com		**College**	cos	
Module Leader	*.Jumana Waleed Salih*		**e-mail**	jumanawaleed@uodiyala.edu.iq	
Module Leader's Acad. Title	Assistant Prof.		**Module Leader's Qualification**	Ph.D.	
Module Tutor	Name (if available)		**e-mail**	E-mail	
Peer Reviewer Name	Name		**e-mail**	E-mail	
Scientific Committee Approval Date	04/08/2024		**Version Number**	1.0	

Relation with other Modules			
Prerequisite module	COM-122	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	This course emphasizes computability and computational complexity theory. Topics include regular and context-free languages, decidable and undecidable problems, reducibility, recursive function theory, time and space measures on computation, completeness, hierarchy theorems, inherently complex problems, oracles, probabilistic computation, and interactive proof systems. • Improve their mathematical thinking skill and habits, including thinking precisely about definitions, stating assumptions carefully, critically reading arguments, and being able to write convincingly. • Be able to understand both finite and infinite formal models of computation and to reason about what they can and cannot compute. • Understand both intuitively and formally what makes some problems either impossible or too expensive to solve with a computer, and what can be done in practice when an unsolvable or intractable problem is encountered. • Reason formally about the cost of computation, and be able to prove useful bounds on the costs of solving problems, including showing that certain problems cannot be solved efficiently. • Learn about some interesting aspects of theoretical computer science, and why understanding them matters even if you are only interested in building practical computing systems.
Module Learning Outcomes	1- Acquisition of the basic concepts of probability and statistical inference. 2- Knowledge and understanding of basic statistical calculations and the software tools used for them. 3- The ability to identify the elements making up a univariate statistical model applied to real situations. 4- The ability to use standard statistical packages and to correctly interpret the lists produced.
Indicative Contents	We explore these questions by developing abstract models of computing machines and reasoning about what they can and cannot compute efficiently. A main goal of this course is for you to understand how theoretical computer scientists reason about these questions, and connecting that theory to practical questions about computing. We will also look at some applications in cryptography that take advantage of problems being hard to solve, and what can be done when a problem cannot be solved or is too expensive to solve.

Learning and Teaching Strategies	
Strategies	• Automata and Language Theory (2 weeks) ○ Finite automata, regular expressions, push-down automata, context-free grammars, pumping lemmas.

	• Computability Theory (3 weeks)
	  ○ Turing machines, the Church-Turing thesis, decidability, the halting problem, reducibility, the recursion theorem.
	• Complexity Theory (7 weeks)
	  ○ Time and space measures of complexity, complexity classes P, NP, L, NL, PSPACE, BPP and IP, complete problems, the P versus NP conjecture, quantifiers and games, hierarchy theorems, provably hard problems, relativized computation and oracles, probabilistic computation, interactive proof systems

Student Workload (SWL) **15 weeks**			
Structured SWL (h/sem)	48	**Structured SWL (h/w)**	3.2
Unstructured SWL (h/sem)	52	**Unstructured SWL (h/w)**	3.4
Total SWL (h/sem)	**100**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	20% (20)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
	Material Covered
Week 1	Introduction, Finite Automata, Regular Expressions
Week 2	Nondeterminism, Closure Properties, Regular Expressions → Finite Automata
Week 3	The Regular Pumping Lemma, Finite Automata → Regular Expressions, CFGs
Week 4	Pushdown Automata, CFG ↔ PDA
Week 5	The CF Pumping Lemma, Turing Machines
Week 6	TM Variants, the Church-Turing Thesis
Week 7	Decision Problems for Automata and Grammars
Week 8	**Midterm Exam**
Week 9	P and NP, SAT, Poly-time Reducibility
Week 10	NP-Completeness
Week 11	Cook-Levin Theorem
Week 12	Space Complexity, PSPACE, Savitch's Theorem
Week 13	PSPACE-Completeness
Week 14	Games, Generalized Geography
Week 15	L and NL, NL = coNL
Week 16	**Preparatory week before the final Exam**

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

Learning and Teaching Resources		
	Text	**Available in the Library?**

Required Texts	Sipser, Michael. *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, 2012. ISBN: 9781133187790.	Yes
Recommended Texts	**Introduction to Probability and Statistics 15th Edition** by William Mendenhall (Author), Robert J. Beaver (Author), Barbara M. Beaver (Author)	no
Websites	https://www.coursera.org/courses?query=theory%20of%20computation	

Grading Scheme			
Group	**Grade**	**Marks %**	**Definition**
Success Group (50 - 100)	**A -** Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information			
Module Title	**Introduction to Object Oriented Language**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	**COM-211**		
ECTS Credits	7		
SWL (hr/sem)	175		
Module Level	2	**Semester of Delivery**	3
Administering Department	com	**College**	cos
Module Leader	Ismael Salih Aref	**e-mail**	asmaelsalih@uodiyala.edu.iq
Module Leader's Acad. Title	Assist.Lect	**Module Leader's Qualification**	MSC
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/08/2024	**Version Number**	1.0

Relation with other Modules			
Prerequisite module	**Programming Language1**	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	The educational objectives of this course are **1- Understanding Core Concepts of OOP:** • **Classes and Objects:** Understanding the foundational building blocks of OOP, including how to define classes (blueprints) and create objects (instances of classes).

	• **Encapsulation:** Learning how to bundle data (attributes) and methods (functions) that operate on the data into a single unit or class, promoting data hiding and reducing complexity. • **Inheritance:** Grasping how new classes can be derived from existing ones, allowing for code reuse and the creation of hierarchical class structures. • **Polymorphism:** Understanding how different classes can be treated as instances of the same class through interfaces, allowing for flexibility in code through method overriding and overloading. • **Abstraction:** Learning to focus on essential qualities of an object while hiding unnecessary details, making complex systems easier to manage. **2- Developing Problem-Solving Skills:** • **Modeling Real-World Systems:** Teaching students to represent real-world entities as objects, helping to develop systems that are intuitive and maintainable. • **Design Patterns:** Introducing common design patterns that solve recurring problems in OOP, fostering best practices in software development. • **Code Reusability:** Emphasizing the importance of creating reusable, modular code that can be easily extended and maintained. **3- Improving Software Design and Architecture:** • **Software Design Principles:** Educating students on principles like SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) to create well-structured and robust code. • **Object-Oriented Analysis and Design (OOAD):** Training students to analyze and design software systems using OOP principles, focusing on creating scalable and maintainable architectures.
Module Learning Outcomes	**1. Knowledge and Understanding:** • **MLO 1:** Demonstrate a comprehensive understanding of the fundamental principles of Object-Oriented Programming, including concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction. • **MLO 2:** Understand and apply the principles of software design and architecture, including the use of design patterns and best practices in OOP. • **MLO 3:** Explain the benefits and limitations of the OOP paradigm in software development, including its impact on code reusability, maintainability, and scalability. **2. Cognitive/Intellectual Skills:** • **MLO 4:** Analyze real-world problems and design effective OOP solutions by modeling appropriate classes, objects, and relationships. • **MLO 5:** Critically evaluate and apply design patterns to solve common software design problems. • **MLO 6:** Assess the trade-offs between different object-oriented designs

	in terms of efficiency, complexity, and scalability. **3. Practical/Professional Skills:** • **MLO 7:** Develop and implement object-oriented software using a relevant programming language (e.g., Java, C++, Python) that adheres to industry standards and best practices. • **MLO 8:** Apply techniques for debugging, testing, and maintaining object-oriented code, including the use of unit tests and version control systems. • **MLO 9:** Work collaboratively in a team environment to design and develop a substantial object-oriented software project, demonstrating effective communication and project management skills. **4. Key Transferable Skills:** • **MLO 10:** Demonstrate problem-solving skills by breaking down complex problems into manageable components using OOP techniques. • **MLO 11:** Communicate technical information effectively, both verbally and in writing, through documentation, code comments, and presentations. • **MLO 12:** Adapt to new and emerging technologies in object-oriented programming, demonstrating lifelong learning and the ability to stay current with industry trends.
Indicative Contents	The indicative content of an Object-Oriented Programming (OOP) course includes an introduction to core concepts like classes, objects, inheritance, encapsulation, polymorphism, and abstraction, along with advanced topics such as composition vs. inheritance, design patterns, and SOLID principles. It also covers object-oriented analysis and design (OOAD), practical implementation in a chosen programming language, and testing/debugging techniques. Students will work on hands-on projects, including collaborative team development, integrating OOP with databases, and exploring modern frameworks and libraries. The course concludes with discussions on contemporary OOP languages, emerging trends, and the future direction of software development.

Learning and Teaching Strategies

Strategies	• Lectures • Tutorials • Problem solving • Lab • Case study • Small project

Student Workload (SWL)

Structured SWL (h/sem)	99	**Structured SWL (h/w)**	6.6
Unstructured SWL (h/sem)	76	**Unstructured SWL (h/w)**	5
Total SWL (h/sem)	**175**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to OOP
Weeks 2	Classes and Objects: Explain Structure of Simple Class
Week 3	Access Specifiers: public, private and protected
Weeks 4	Encapsulation principle (data hiding)
Week 5	Constructor (usage and advantage)
Week 6	Destructor (Purpose, syntax, advantages)
Week 7	Mid-term Exam
Week 8	Inheritance Basics
Weeks 9	Types of Inheritance
Weeks 10	Function Overriding in class
Weeks 11	constructor overloading
Weeks 12	Polymorphism
Week 13	Virtual Functions
Week 14	Operator Overloading
Week 15	Homework Sheets solving
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Simple Cass Structure

Week 2	Create Classes and objects
Week 3	Public and private example (control data access)
Week 4	Protected access specifier and private difference by examples.
Week 5	Structures of contractor
Week 6	Pointer and destructor roles
Week 7	Mid-term Exam
Week 8	Explain Inheritance structure by examples
Week 9	Many examples to explain inheritance levels and types.
Week 10	Apply Function Overriding in class
Week 11	Apply constructor overloading with different arguments
Week 12	Explain Polymorphism concepts by examples.
Week 13	Examples to show usage of Virtual Functions
Week 14	Operator Overloading (unary and binary)
Week 15	Solving Examples Sheet
Week 16	Exam

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	• Programming in C++ Frank Vahid and Roman Lysecky Available through the zyBooks website directly • A C++ compiler and/or IDE. There are many out there, but the only two that are officially supported: - CLion (on Windows and macOS) - Visual Studio (Windows only)	Yes
Recommended Texts	▪ Think Like a Programmer, An Introduction to Creative Problem Solving V. Anton Spraul ISBN: 978-1593274245 ▪ A good text editor, such as: Notepad++ (This is my personal favorite) Sublime Text Atom, or Vim, or anything else you might prefer	No
Websites	1-http://www.cplusplus.com/ 2-https://www.youtube.com/@IsmaelSalih	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors

	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Data Structure**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-212**		
ECTS Credits	6		
SWL (hr/sem)	**150**		
Module Level	2	**Semester of Delivery**	3
Administering Department	Type Dept. Code	**College**	Type College Code
Module Leader	Ali Abdulrahman Mahmood	**e-mail**	alialani@uodiyala.edu.iq
Module Leader's Acad. Title	lecturer assistant	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/8//2024	**Version Number**	1.0

Relation with other Modules

Prerequisite module	COM-121	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	1. The objective required of the student in order to successfully pass the course requirements is to understand how - through - data is represented and stored - inside the calculator. 2. The student's realization of the types of algorithms used in data representation. 3. Developing the student's ability to use the software available in this field, in addition to the skills he acquires in data processing and how they are represented inside the calculator. 4. Developing the student's ability to write software that handles data and how to represent it in a calculator.
Module Learning Outcomes	Important: Write at least 6 Learning Outcomes, better to be equal to the number of study weeks. 1. Display an Introduction to Data structures. 2. List the Strategies for choosing the right data structure. 3. Explain the first Linear data structures- Array. 4. Explain the Pointer and its operations,applications. 5. Describe the Structure. 6. Define the Linked list and its operations,applications. 7. Identify the second Linear Data Structure: The Stack 8. Discuss the third Linear Data Structure: The Queue 9. Explain the The Graph 10. Explain the The Tree. 11. Identify the Heaps and its operations 12. Describe the searching, sorting
Indicative Contents	Indicative content includes the following. Basics • Algorithm Specifications: Performance Analysis and Measurement (Time and space analysis of algorithms- Average, best and worst case analysis) Introduction To Data Structure: • Data Management concepts, • Data types – primitive and non-primitive, • Types of Data Structures- Linear & Non Linear Data Structures. Linear Data Structure • Array: Representation of arrays, Applications of arrays, sparse matrix and its representation., • Stack: Stack-Definitions & Concepts, Operations On Stacks, Applications of Stacks, Polish Expression, Reverse Polish Expression And Their Compilation, Recursion, Tower of Hanoi, • Queue: Representation Of Queue, Operations On Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue, • Linked List: Singly Linked List, Doubly Linked list, Circular linked list ,Linked implementation of Stack, Linked implementation of Queue, Applications of linked list. Nonlinear Data Structure : • Tree-Definitions and Concepts, Representation of binary tree, Binary tree traversal (Inorder, postorder, preorder), • Threaded binary tree, • Binary search trees, • Conversion of General Trees To Binary Trees, • Applications Of Trees- Some balanced tree mechanism, eg. AVL trees, 2-3 trees, Height Balanced, Weight Balance , • Graph-Matrix Representation Of Graphs, Elementary Graph operations,(Breadth First Search, Depth First Search, Spanning Trees, Shortest path, Minimal spanning tree). SORTING And SEARCHING • Insertion Sort, • Quick Sort,

	• Merge Sort, • Heap Sort, • Sorting On Several Keys, • List and Table Sort,

Learning and Teaching Strategies

Strategies	• At the start of course, the course delivery pattern, prerequisite of the subject will be discussed. • Lectures will be conducted with the aid of multi-media projector, black board, OHP etc. • Attendance is compulsory in lecture which carries 10 marks in overall evaluation. • One internal exam will be conducted as a part of internal theory evaluation. • Assignments based on the course content will be given to the students for each unit and will be evaluated at regular interval evaluation. • Surprise tests/Quizzes/Seminar/tutorial will be conducted having a share of five marks in the overall internal evaluation. • The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures. • Experiments shall be performed in the laboratory related to course contents.

Student Workload (SWL)

Structured SWL (h/sem)	79	**Structured SWL (h/w)**	5.2
Unstructured SWL (h/sem)	71	**Unstructured SWL (h/w)**	4.7
Total SWL (h/sem)	**150**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	4 and 8	LO #1, #2 and #3
	Assignments	2	10% (10)	6 and 14	LO #3, #4 and #6, #7
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	15	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered

Week 1	an Introduction to Data structures.
Week 2	the Strategies for choosing the right data structure.
Week 3	the Linear data structures- Array /one dimensional array
Week 4	An array/ two dimensional array
Week 5	the Pointer and its operations, applications
Week 6	the Structure.
Week 7	Mid- term Exam
Week 8	the Linked list, linked list types and its operations, applications
Week 9	the second Linear Data Structure: The Stack
Week 10	the third Linear Data Structure: The Queue
Week 11	The Graph
Week 12	The Tree, binary tree
Week 13	the Heaps and its operations
Week 14	Sorting
Week 15	Searching , Searching types
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Introduction to C++ Programming Language.
Week 2	1D array operations
Week 3	2D array operations
Week 4	structures & pointers in C
Week 5	Write a program for linked list insertion, deletion & copy
Week 6	Stack operations Write a program to perform PUSH, POP, PEEP & CHANGE operations on Stack using array.
Week 7	Stack operations Write a program to perform PUSH, POP, PEEP & CHANGE operations on Stack using linked list.
Week 8	Midterm Exam
Week 9	Queue Operations Write a program to implement insertion & deletion in a queue using array.
Week 10	Queue Operations Write a program to implement insertion & deletion in a queue using linked list.
Week 11	Circular Queue Operations Write a program to implement insertion & deletion in a circular
Week 12	Sorting and searching: Write a program to perform • Selection sort
Week 13	Sorting and searching: Write a program to perform • Merge sort
Week 14	Sorting and searching: Write a program to perform • Quick sort
Week 15	Preparatory week before the final Exam

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	1-Tanenbaum Aaron M, Langsam Yedidyah, Augenstein J Moshe, Data Structures using C. 2-Tremblay J.P and Sorenson P.G, An introduction to data structures with applications, Tata McGraw Hill, 2nd Edition	Yes
Recommended Texts	1-Fundamentals of Computer Algorithms by Horowitz, Sahni,Galgotia Pub. 2001 ed. 4. 2-Fundamentals of Data Structures in C++-By Sartaj Sahani. 3- Data Structures: A Pseudo-code approach with C -By Gilberg & Forouzan Publisher-Thomson Learning	yes
Websites	Data Structures by Lipschutz Seymour [Schaum's Outline]	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.

Module Information

Module Title	**Numerical Methods**	**Module Delivery**	
Module Type	**Basic**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	**SCI-103**		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	2	**Semester of Delivery**	3

Administering Department	com	**College**	science
Module Leader	Iraq ali hussein	**e-mail**	iraqali@uodiyala.edu.iq
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	24/08/2024	**Version Number**	1.0

Relation with other Modules			
Prerequisite module	**Computational mathematics**	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	The main objectives of the course are to: ● Provide an understanding of numerical methods for solving mathematical problems. ● Develop skills in implementing numerical algorithms. ● Introduce the application of numerical methods to real-world engineering problems. ● Focus on error analysis and computational stability of algorithms. ● Teach students how to develop and analyze their own numerical algorithms.
Module Learning Outcomes	After completion of the course students are expected to be able to: 1. Understand the fundamental principles of numerical methods. 2. Analyze the accuracy and stability of numerical solutions. 3. Implement numerical algorithms in a programming environment. 4. Apply numerical methods to solve practical engineering and scientific problems. 5. Critically evaluate the performance of different numerical methods.
Indicative Contents	1. Introduction to Numerical Methods and Error Analysis. 2. Solutions of Nonlinear Equations. 3. Numerical Differentiation and Integration. 4. Numerical Solutions of Ordinary Differential Equations. 5. Numerical Linear Algebra. 6. Interpolation and Curve Fitting.

	7. Optimization Techniques. 8. Stability and Convergence of Algorithms.

Learning and Teaching Strategies

Strategies	• Lectures, Lab Tutorials, Practical Exercises, and Assignments

Student Workload (SWL)

Structured SWL (h/sem)	66	**Structured SWL (h/w)**	4.4
Unstructured SWL (h/sem)	48	**Unstructured SWL (h/w)**	3.2
Total SWL (h/sem)	**125**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	2,6 and 9,11	LO #2, #6 and #9, #11
	Assignments	2	10% (10)	3,5 and 10,12	LO #3, #5 and #10, #12
	Home Works	1	10% (10)	2,5 and 8,11	LO #2, #5 and #8, #11
	Report	1	10% (10)	13	LO #13
	Projects / Lab.	2	10% (10)	Continuous	All
Summative assessment	**Midterm Exam**	2hr	10% (10)	8	LO #8
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to Numerical Methods * Definitions and classifications * Importance and applications * Overview of error analysis
Week 2	Solutions of Nonlinear Equations * Bisection method * Newton-Raphson method * Secant method

Week 3	Numerical Differentiation and Integration * Trapezoidal rule * Simpson's rule * Numerical differentiation formulas
Week 4	Numerical Solutions of ODEs * Euler's method * Runge-Kutta methods * Stability of ODE solutions
Week 5	Numerical Linear Algebra * Gaussian elimination * LU decomposition * Iterative methods
Week 6	Interpolation * Lagrange interpolation * Newton's divided difference * Spline interpolation
Week 7	Midterm Exam
Week 8	Optimization Techniques * Unconstrained optimization * Constrained optimization * Applications in engineering
Week 9	Stability and Convergence * Analysis of numerical algorithms * Convergence criteria * Practical examples
Week 10	Case Studies and Applications * Application of numerical methods in engineering * Discussion of case studies
Week 11	Introduction to Numerical Methods * Definitions and classifications * Importance and applications * Overview of error analysis
Week 12	Solutions of Nonlinear Equations * Bisection method * Newton-Raphson method * Secant method
Week 13	Numerical Differentiation and Integration * Trapezoidal rule * Simpson's rule * Numerical differentiation formulas
Week 14	Numerical Solutions of ODEs * Euler's method * Runge-Kutta methods * Stability of ODE solutions
Week 15	Numerical Linear Algebra * Gaussian elimination * LU decomposition * Iterative methods

Delivery Plan (Weekly Lab. Syllabus)	
	Material Covered

Weeks 1 and 2,3	Introduction to Numerical Software * Overview of software tools used for numerical analysis * Setting up the environment * Basic operations
Weeks 4 and 5,6	Nonlinear Equations * Implementing bisection method * Implementing Newton-Raphson method * Comparison of methods
Weeks 7,8 and 9	Numerical Integration * Implementing trapezoidal and Simpson's rule * Numerical differentiation * Error analysis
Weeks 9 and10 ,11	Ordinary Differential Equations * Implementing Euler's method * Implementing Runge-Kutta methods * Solving real-world problems
Weeks 12,14 and 14	Linear Algebra * Implementing Gaussian elimination * LU decomposition * Solving systems of linear equations

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	● Burden, R. L., & Faires, J. D. (2011). Numerical Analysis (9th Edition). Cengage Learning. ● Chapra, S. C., & Canale, R. P. (2015). Numerical Methods for Engineers (7th Edition).	
Recommended Texts	● Sauer, T. (2012). Numerical Analysis. Pearson. ● Atkinson, K. E. (1989). An Introduction to Numerical Analysis (2nd Edition). Wiley.	
Websites		

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A -** Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Web applications Development**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-214**		
ECTS Credits	6		
SWL (hr/sem)	150		
Module Level	2	**Semester of Delivery**	3
Administering Department	com	**College**	cos
Module Leader	Dheyab Salman Ibrahim	**e-mail**	dr.dheyab@uodiyala.edu.iq
Module Leader's Acad. Title	Assistant Prof	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/08/2024	**Version Number**	1.0

Relation with other Modules

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

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	Web Development course syllabus aims to teach about front-end, back-end, and full-stack Web Development. Web Development course covers various topics under Web Development such as Database Management, Web Publishing, Web Design, and Web Programming. . This is an online course. All official course materials will be made available online at https://gitlab.msu.edu/cse477-spring-2022. This course does not use desire2learn. If you can not access the course repository, please contact a member of the course staff. The contents of the course require a login using an MSU ID and password.
Module Learning Outcomes	This course provides an overview of contemporary techniques, and tools used for web application development. More specifically, this course covers the three essential technology components of web applications (frontend, backend, and databases), as well as the internet technologies used to host, distribute, and scale web applications. A list of key topics include: How the internet works, internet protocols, domain name servers, web hosting, HTML, CSS, Javascript, conventions and best practices, DOM manipulation, Python Flask, sessions, cookies, relational databases, NoSQL Databases, database optimization, containerization, version control systems, APIs and Microservices. Objective Students completing this course are expected to be able to: Understand the unique aspects of web application design. Work in resource sensitive and resolution variant environments. Apply common patterns in web development
Indicative Contents	Web development is building websites and web applications like Facebook, Twitter, or internal web portals within businesses. Web development has two disciplines: front-end and back-end. **Front-end** is visual and interactive aspects of a website. You will learn HTML, CSS and JavaScript to master front-end web development. **Back-end** is all the logic behind the scenes that supports your website: databases, user management, etc. You will need to learn any **one** of the following back-end languages and frameworks:

Learning and Teaching Strategies	
Strategies	Web design is foundational material for computer science: Many areas of computer science require the ability to work with concepts from web design technologes, specifically material from such HTML, CSS, and Java Script. The main strategy that will be adopted in delivering the Web design structures module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. The module will include a combination of classes, an interactive tutorials.

Student Workload (SWL)

Structured SWL (h/sem)	79	**Structured SWL (h/w)**	5.1
Unstructured SWL (h/sem)	71	**Unstructured SWL (h/w)**	4.7
Total SWL (h/sem)	**150**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to Web Development
Week 2	HyperText Markup Language and Cascading Style Sheets
Week 3	JS Functions and Objects
Week 4	JavaScript and HTTP (forms)
Week 5	Database Interaction & UI
Week 6	Mathematical Structure for Computer Science
Week 7	Front End Libraries & Frameworks
Week 8	Exam
Week 9	Intro to Programming with the use of JavaScript
Week 10	Website Structure and Hosting
Week 11	Creating stylish Websites
Week 12	Creating HTML Forms
Week 13	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)	
	Material Covered
Week 1	Introduction to HTML
Week 2	Introducing Flexbox
Week 3	JavaScript for Front-end
Week 4	Syntax & Creating Concepts with JavaScript
Week 5	Design Patterns & Object Modelling
Week 6	HTTP Requests & Routes
Week 7	Intro to Build Tools
Week8	Express Framework, Building scalable web applications
Week9	OOPS
Week10	JSON & AJAX
Week11	Development Environment & Tools

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	- -How the Internet Works, Preston Gralla, Pearson Education, Eighth Edition. -Internet for Everyone, Alexis Leon, S. Chand (G/L) & Company Ltd; Second Edition.	Yes
Recommended Texts	- -DATA COMMUNICATIONS AND NETWORKING, Fourth Edition, Behrouz A. Forouzan - "Web Programming Step by Step" by Marty Stepp, Jessica Miller, Victoria Kirst. -A useful web site for learning more about web development - Available here: https://www.geeksforgeeks.org/web-1-0-web-2-0-and-web-3-0-with-their-difference/ - (1970) https://websitebuilders.com/how-to/glossary/web2/ - Available here: https://askanydifference.com/difference-between-web-1-0-and-web-2-0-with-table/	Yes
Websites	http://www.w3schools.com Tutorial points Simply easy learining	

Grading Scheme			
Group	**Grade**	**Marks %**	**Definition**
Success Group	**A -** Excellent	90 - 100	Outstanding Performance

(50 - 100)	**B** - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information			
Module Title	**Computer Architecture**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	**COM-223**		
ECTS Credits	4		
SWL (hr/sem)	100		
Module Level	2	**Semester of Delivery**	4
Administering Department	com	**College**	cos
Module Leader	Khalid Mohammed Saffer	**e-mail**	dr.khaledmoh@uodiyala.edu.iq
Module Leader's Acad. Title	Assistant Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	20/08/2024	**Version Number**	1.0

Relation with other Modules			
Prerequisite module	**Logic Design** **COM-123**	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	In a Computer Architecture module, the objectives typically revolve around understanding the structure, operation, and design of computer systems. Here are some common objectives you might encounter: 1. Understanding Fundamental Concepts. 2. Exploring Processor Design. 3. Memory Hierarchy and Management. 4. Understanding Data Paths and Control. 5. Performance Analysis. 6. Exploring Parallelism. 7. Input/Output Systems. 8. Pipelining and Hazards. 9. Exploring Emerging Trends. 10. Practical Application.
Module Learning Outcomes	Learning outcomes for a Computer Architecture module define what students should be able to do upon successful completion of the course. Here are some typical learning outcomes: 1. Understand and Describe Core Concepts. 2. Analyze and Design Processor Architectures. 3. Evaluate and Optimize System Performance. 4. Understand Memory Organization and Hierarchy. 5. Demonstrate Knowledge of Parallelism. 6. Implement and Simulate Basic Computer Systems. 7. Identify and Address Pipelining Issues. 8. Understand Input/Output Mechanisms. 9. Apply Knowledge to Emerging Technologies. 10. Collaborate and Communicate Effectively.
Indicative Contents	Introduction to computer architecture, basic organization of computer, basic operational concept, bus structures, requirements of I/O system, I/O interfacing techniques, memory system organization, 8085 Microprocessor, Intel core family.

Learning and Teaching Strategies

Strategies	• lecture • Tutorial • Conducting discussion panels within the lecture • Giving weekly homework • Asking questions during the lecture

Student Workload (SWL)

Structured SWL (h/sem)	48	**Structured SWL (h/w)**	3.2
Unstructured SWL (h/sem)	52	**Unstructured SWL (h/w)**	3.4
Total SWL (h/sem)	**100**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	4	10% (10)	2,6 and 9,11	LO #2, #6 and #9, #11
	Assignments	2	10% (10)	3,5 and 10,12	LO #3, #5 and #10, #12
	Home Works	1	10% (10)	2,5 and 8,11	LO #2, #5 and #8, #11
	Report	1	10% (10)	13	LO #13
Summative assessment	**Midterm Exam**	2hr	10% (10)	8	LO #8
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to computer architecture
Week 2	Basic organization of computer
Week 3	Basic operational concept
Week 4	Bus structures
Week 5	Requirements of I/O system
Week 6	I/O interfacing techniques
Week 7	Memory system organization

Week 8	Midterm exam
Week 9	Memory hierarchy
Week 10	Memory structure and its requirement
Week 11	Associative memory
Week 12	Cache memory
Week 13	8085 Microprocessor
Week 14	Intel core family
Week 15	Intel core family

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Weeks 1 and 2	
Weeks 3 and 4	
Weeks 5,6 and 7	
Weeks 8 and 9	
Weeks 10,11 and 12	
Weeks 13 and 14	
Week 15	

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	Mano, M. Morris, Computer System Architecture, 3rd Edition, Prentice-Hall, Inc., 1993.	
Recommended Texts	- Mostafa Abd-El-Barr, Hesham El-Rewini, "Fundamentals of Computer Organization and Architecture", A John Wiley & Sons, Inc Publication, 2005. - M. Morris Mano, Computer Engineering Hardware Design, 1st Edition, Prentice-Hall, Inc., 1988.	
Websites		

Grading Scheme

Group	**Grade**	**Marks %**	**Definition**

Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Computer Graphics**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-222**		
ECTS Credits	**4**		
SWL (hr/sem)	**100**		
Module Level	2	**Semester of Delivery**	4
Administering Department	Type Dept. Code	**College**	Type College Code
Module Leader	Juliet Kadum Dawood	**e-mail**	julietkadum@uodiyala.edu.iq
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	M.sc.
Module Tutor		**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	07/08/2024	**Version Number**	1.0

Relation with other Modules

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

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	1- The main objective of this module is to introduce to the students the concepts of computer graphics. 2- Defining the important steps in designing geometric shapes 3- Studying the methods of mathematical representation of geometric shapes and geometri transformations. 4-Preparing qualified graduates to work in the field of computerized applications by representing data in the form of computer graphics. 5- Introducing the student to the field of computer graphics and processing visual and engineering information using computer technologies. 6- focuses on the mathematical and computational foundations of image generation and processing.
Module Learning Outcomes	Important: Write at least 6 Learning Outcomes, better to be equal to the number of study weeks. 1- Identify the basic elements of graphics and their applications. 2- understand of the structure of an interactive computer graphics system. 3- List the various terms associated with graphics mode . 4- studying all the various of algorithms associated with Drawing lines. 5- studying all the various of algorithms associated with Drawing circle. 6- Have a knowledge and understanding of geometrical transformations and 2D viewing. 7- Have a knowledge and understanding of techniques for representing 2D geometrical objects. 8- Have a knowledge and understanding of geometrical transformations and 3D viewing. 9- Have a knowledge and understanding of techniques for representing 3D geometrical objects. 10- Have a knowledge and understanding of interaction techniques. 11- Explain all(functions) related with drawing (pixel,line,circle..). 12- Be able to create interactive graphics applications. 13- Perform simple 2D graphics with lines, curves and can implement algorithms to rasterizing simple shapes, fill and clip polygons. 14- studying all the various of algorithms associated with clipping. 15- Explain the windowing and viewing. 16- Identify the applications of windowing and viewing.
Indicative Contents	This course introduces computer Graphics . This course has been approved to satisfy the Comprehensive Articulation Agreement for transferability as a pre-major .

	Home Works and Assignments Attendance is mandatory. Every class is important. All deadlines are hard. Under normal circumstances late work will not be accepted. Students are required to take all the tests. No make-up tests will be given under normal circumstances. Any form of cheating on exams/assignments/quizzes is subject to serious penalty Attendance 75% attendance is mandatory. Latecomers will be marked as absent. Evaluation Criteria Assignments/projects 20% Quizzes 10% Mid-Term 20% Final 50%

Learning and Teaching Strategies

Strategies	The main strategy that will be adopted in delivering this module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. This will be achieved through classes, interactive tutorials and by considering types of simple experiments involving some that are interesting to the students.

Student Workload (SWL)

Structured SWL (h/sem)	63	**Structured SWL (h/w)**	4.2
Unstructured SWL (h/sem)	37	**Unstructured SWL (h/w)**	1.8
Total SWL (h/sem)	**100**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction - Computer graphics
Week 2	Basics of (Elementary Figures Plotting Points)
Week 3	Line Drawing Horizontal and Vertical Lines,DDA algorithm
Week 4	Arbitrary Lines, Bresenham's Line Algorithm
Week 5	Functions to draw line, some drawing related with line
Week 6	Circle Drawing(Functions to draw circle, some drawing related with circle)
Week 7	**Mid-term Exam**
Week 8	Bresenham's Circle Algorithm
Week 9	Introduction- Geometric Transformations(Translate,Rotate,Reflection,scaling)
Week 10	Two-Dimensional Transformations
Week 11	Clipping and Windowing
Week 12	Windowing and viewing
Week 13	Introduction -Three-Dimensional (3-D)
Week 14	(3-D) Three Dimensional Transformations(Translate,Rotate,Reflection,scaling)
Week 15	Projections, Parallel Orthographic Projection
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Lab 1: Introduction to c++
Week 2	Lab 2: application to algorithm (Line)
Week 3	Lab 3: application to algorithm (circle)
Week 4	Lab 4: application to algorithm (Two-Dimensional (2-D))
Week 5	Lab 5: application to algorithm (Three-Dimensional (3-D))
Week 6	Lab 6: application to algorithm (clipping and windowing and viewing)
Week 7	Lab 7: application to algorithm (Projections)

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	1- "Principles of Interactive Computer Graphics", William M. Newman and Robert F. Sproull, McGraw-Hill International Book Company, 1984. 2- "Computer Graphics with Pascal", Marc Berger, the Benjamin / Cummings Publishing Company, 1986. 3- "Computer Graphics",Zhigang Xiang and Roy A. Plastock, Schaum's outline Series, McGraw-Hill Company, 1992. 4- "Computer Graphics C Version", Donald Hearn and M. Pauline Baker, Prentice-Hall Company, 1997.	Yes

Recommended Texts	1-"FUNDAMENTALS OF COMPUTER GRAPHICS USING MATLAB LANGUAGE", Amaal Kadum Dawood&Juleet Kadim Daood&Jinan Redha Mutar,2022	Yes
Websites	http://www.edm2.com/0507/introcpp1.html	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Algorithm Design and Analysis**	**Module Delivery**	
Module Type	**C**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	COM-221		
ECTS Credits	**5**		
SWL (hr/sem)	**125**		
Module Level	2	**Semester of Delivery**	4
Administering Department	COM	**College**	COS
Module Leader	Name : Ahmed Khudhair Abbas	**e-mail**	Dr.ahmed.k.abbas@uodiyala.edu.iq
Module Leader's Acad. Title	Assist. Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor		**e-mail**	
Peer Reviewer Name		**e-mail**	
Scientific Committee Approval Date	02/08/2024	**Version Number**	1.0

Relation with other Modules

Prerequisite module	COM-212 Data Structurs	**Semester**	3
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives

Students will explore and become more familiar with:

1. Certainly! Here are ten course objectives commonly found in Algorithms Design and Analysis courses:
2. Understand fundamental algorithmic concepts: Gain a solid understanding of fundamental concepts such as algorithm efficiency, complexity analysis, and problem-solving strategies.
3. Analyze algorithm efficiency: Learn how to analyze the time and space complexity of algorithms, enabling you to make informed decisions about algorithm selection for different problem scenarios.
4. Apply algorithmic problem-solving techniques: Acquire a range of problem-solving techniques, including divide and conquer, dynamic programming, greedy algorithms, and graph algorithms, and apply them to solve real-world problems.
5. Analyze and compare algorithmic strategies: Learn how to analyze and compare different algorithmic strategies, considering factors such as worst-case, average-case, and best-case scenarios.
6. Explore advanced algorithms: Study advanced algorithmic topics such as sorting algorithms (e.g., quicksort, merge sort), searching algorithms (e.g., binary search, hash tables), and graph algorithms (e.g., Dijkstra's algorithm, minimum spanning trees).
7. Apply algorithm analysis techniques: Acquire skills in conducting algorithm analysis through mathematical modeling, empirical analysis, and algorithmic simulations to evaluate algorithm performance.
8. Understand the limits of computation: Gain an understanding of the theoretical limits of computation, including concepts like NP-completeness and the P versus NP problem, and appreciate the challenges associated with solving computationally intractable problems.

These course objectives aim to equip students with a strong foundation in algorithm design and analysis, enabling them to develop efficient algorithms and solve complex computational problems effectively.

Module Learning Outcomes

Upon the completion of the course, students will be able to:

1. Demonstrate a solid understanding of fundamental algorithmic concepts, including algorithm efficiency, complexity analysis, and problem-solving strategies.
2. Analyze and evaluate the efficiency of algorithms, considering factors such as time

	complexity, space complexity, and their impact on problem-solving. 3. Design and implement efficient algorithms to solve a variety of complex problems, optimizing for time and space efficiency. 4. Apply a range of problem-solving techniques, including divide and conquer, dynamic programming, greedy algorithms, and graph algorithms, to solve real-world problems effectively. 5. Evaluate and select appropriate data structures for different algorithmic scenarios, considering their strengths and weaknesses in relation to algorithm design and efficiency. 6. Analyze and compare different algorithmic strategies, considering worst-case, average-case, and best-case scenarios, to make informed decisions about algorithm selection. 7. Understand and implement advanced algorithms such as sorting algorithms, searching algorithms, and graph algorithms, and analyze their performance characteristics. 8. Understand the theoretical limits of computation, including concepts like NP-completeness and the P versus NP problem, and recognize the challenges associated with solving computationally intractable problems. These module learning outcomes reflect the desired knowledge and skills that students should acquire by the end of the course, demonstrating their proficiency in algorithm design, analysis, and problem-solving.
Indicative Contents	**Indicative content includes the following.** 1. An introduction to algorithms 2. Complexity of Algorithm 3. Asymptotic notation 4. Divide-and-conquer 5. Brute-force algorithms 6. Greedy algorithms 7. Graph Algorithms 8. Simple numerical algorithms 9. Hash tables, including collision- avoidance strategies

Learning and Teaching Strategies

Strategies	The class will "meet" the equivalent of two one-hour & fifteen minutes for lecture/discussion each week. The lectures and discussions will be a combination of synchronous and asynchronous discussions using WebCT. Students must have access to the Internet to facilitate demonstrating and using software. Many of the assignments should stress hands-on applications by the students. Each student will be expected to participate in all lectures. Class participation by all is expected.

Student Workload (SWL)

Structured SWL (h/sem)	64	**Structured SWL (h/w)**	4.1
Unstructured SWL (h/sem)	61	**Unstructured SWL (h/w)**	4
Total SWL (h/sem)	**125**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1, 2	**An introduction to algorithms**
Week 3,4	**Complexity of Algorithm**
Week 5	**Asymptotic notation**
Week 6,7	**Divide-and-conquer**
Week 8	**MID EXAM**
Week 9	**Brute-force algorithms**
Week 10,11	**Greedy algorithms**
Week 12,13	**Graph Algorithms**
Week 14	**Simple numerical algorithms**
Week 15	**Hash tables, including collision- avoidance strategies**
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Lab 1: Prime numbers , Factorial of a number
Week 2	Lab 2: Greatest Common Divisor GCD, Fibonacci Series
Week 3	Lab 3: Binary Search
Week 4	Lab 4: Quick Sort Algorithm
Week 5	Lab 5: Selection Sort Algorithm
Week 6	Lab 6: Insertion Sort Algorithm
Week 7	Lab 7: Heap Sort Algorithm
Week 8	Lab 8: Merge Sort Algorithm
Week 9	Lab 9: Bubble Sort Algorithm
Week 10	Lab 10: String matching algorithm
Week 11	Lab 11: coin-change algorithm
Week 12	Lab 12: Kruskal's Algorithm
Week 13	Lab 13: Prim's Algorithm
Week 14	Lab 14: Examples
Week 15	Lab 15: Examples

Learning and Teaching Resources

	Text	Available in the Library?
Required Textbooks	Introduction to Algorithms - *Third Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein*	Yes
Essential References Materials	Design and Analysis of Algorithms A Contemporary Perspective Cambridge University Press April 2019	
Electronic Materials	https://www.geeksforgeeks.org/design-and-analysis-of-algorithms/	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings

	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Introduction to Python**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☒ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	**COM-224**		
ECTS Credits	5		
SWL (hr/sem)	125		
Module Level	2	**Semester of Delivery**	4
Administering Department	com	**College**	cos
Module Leader	Ali Hussein Fadil	**e-mail**	Ali.hussien@uobasrah.edu.iq
Module Leader's Acad. Title		**Module Leader's Qualification**	
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	20/08/2024	**Version Number**	1.0

Relation with other Modules

Prerequisite module	**Introduction to Object Oriented Language**	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	Python Programming is intended for software engineers, systems analysts, program managers and user support personnel who wish to learn the Python programming language.
Module Learning Outcomes	The learning objectives of this course are: ▪ To understand why Python is a useful scripting language for developers. ▪ To learn how to design and program Python applications. ▪ To learn how to use lists, tuples, and dictionaries in Python programs. ▪ To learn how to identify Python object types. ▪ To learn how to use indexing and slicing to access data in Python programs. ▪ To define the structure and components of a Python program. ▪ To learn how to write loops and decision statements in Python. ▪ To learn how to write functions and pass arguments in Python. ▪ To learn how to build and package Python modules for reusability. ▪ To learn how to read and write files in Python. ▪ To learn how to design object-oriented programs with Python classes. ▪ To learn how to use class inheritance in Python for reusability. ▪ To learn how to use exception handling in Python applications for error handling.
Indicative Contents	**Introduction to Programming in Python:** Introduction to Programming in Python: What Is Python? Features of Python, Python environment set up. [5 hrs.] **Download & Install Python:** Download your operating system-compatible Python Interpreter, install Python, customize Python shell, and write & execute Python programs using Interactive mode and script mode. Python PyCharm or IDE sets Python for PyCharm IDE, configures PyCharm IDE, and writes & executes Python programs. [5 hrs.] **Structure of a Python Program Basics of Programming in Python:** Input statement, output statement, variables, operators, numbers, Literals, strings, lists and tuples, dictionaries. [16 hrs.] **Conditionals, Loops, and Functions. Conditionals and Loops:** if statement, else Statement, elif Statement, while Statement, for Statement break Statement, continue Statement, pass Statement. Functions: Built-in Functions, User-defined functions: Defining a Function, Calling a Function, Various Function Arguments. [25 hrs.] **Files, Modules, and Introduction to Advanced Python. Files:** File Objects, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages. [20 hrs.]

Learning and Teaching Strategies

Strategies	• The main strategy that will be adopted in delivering this module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. This will be achieved through classes, interactive tutorials, and by considering types of simple experiments involving some sampling activities that are interesting to the students.

Student Workload (SWL)

Structured SWL (h/sem)	28	**Structured SWL (h/w)**	5.2
Unstructured SWL (h/sem)	71	**Unstructured SWL (h/w)**	4.7
Total SWL (h/sem)	**125**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	3	10% (10)	2,6 and 9,11	LO #2, #6 and #9, #11
	Assignments	1	10% (10)	3,5 and 10,12	LO #3, #5 and #10, #12
	Home Works	2	10% (10)	2,5 and 8,11	LO #2, #5 and #8, #11
	Projects / Lab.	1	10% (10)	Continuous	All
	Report	1	10% (10)	13	LO #13
Summative assessment	**Midterm Exam**	2hr	10% (10)	8	LO #8
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to Python programming and installation, why using Python, python connected environment type Python PyCharm or IDE, set Python for PyCharm IDE, libraries, Python code (print), experiment with example code.

Week 2	Data Types: Int, float, Boolean, string, and list; variables, expressions, statements, precedence of operators, comments.
Week 3	modules, functions--- function and its use, flow of execution, parameters, and arguments. data type, variable types, and experiment function, string type.
Week 4	CONTROL FLOW, LOOPS Conditionals: Boolean values and operators, conditional (if), alternative (if-else).
Week 5	CONTROL FLOW, chained conditional (if-elif-else).
Week 6	LOOPS Conditionals: Iteration: while, for, break, continue.
Week 7	**Mid Exam**
Week 8	FUNCTIONS, ARRAYS Fruitful functions: return values, parameters, local and global scope, and function composition.
Week 9	Recursion.
Week 10	Strings: string slices, immutability, string functions and methods, string module
Week 11	Python arrays, Access the Elements of an Array, array methods.
Week 12	LISTS, TUPLES, DICTIONARIES Lists: list operations, list slices, list methods, list loop
Week 13	Mutability, aliasing, cloning lists, list parameters, list comprehension;
Week 14	Tuples: tuple assignment, tuple as return value, tuple comprehension; Dictionaries: operations and methods, comprehension
Week 15	FILES, EXCEPTIONS: Files and exceptions: text files, reading and writing files.

Delivery Plan (Weekly Lab. Syllabus)	
	Material Covered
Week 1	Introduction to Python programming and installation Python environment, PyCharm.
Week 2	Example of first Python code (print) code, string type.
Week 3	An example of Python data type and variables is a library of PyCharm
Week 4	Starting of Python project.
Week 5	Example of Python mathematic operations, List, and operation
Week 6	Example of Python list.

Week 7	More examples of list function
Week 8	Apply of python project 1.
Week 9	Example of Python Tubes
Week 10	Example of Python Tubes
Week 11	Example of Python function
Week 12	Apply of python project 2.
Week 13	Example of Conditional statements (if-else).
Week 14	Example of Loops (for, while, do-while).
Week 15	First Dissection of Python Project.

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	Think Python: How to Think Like a Computer Scientist", Allen B. Downey 2nd edition, Updated for Python 3, Shroff/O 'Reilly Publishers, 2016.	
Recommended Texts	"Core Python Programming", W.Chun, Pearson, Kenneth A. Lambert, Cengage, Learning Python, Mark Lutz, Orielly.	
Websites	https://www.gcreddy.com/2021/05/python-programming-syllabus.html.	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**OOP Programming Fundamentals**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	COM-211		
ECTS Credits	4		
SWL (hr/sem)	100		
Module Level	1	**Semester of Delivery**	2
Administering Department	com	**College**	cos
Module Leader	Ismael Salih Aref	**e-mail**	asmaelsalih@uodiyala.edu.iq
Module Leader's Acad. Title	Assist.Lect	**Module Leader's Qualification**	MSC
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/07/2024	**Version Number**	1.0

Relation with other Modules

Prerequisite module	**Programming Language c++**	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	The educational objectives of this course are **4- Understanding Core Concepts of OOP:** • **Classes and Objects:** Understanding the foundational building blocks of OOP, including how to define classes (blueprints) and create objects (instances of classes).

	• **Encapsulation:** Learning how to bundle data (attributes) and methods (functions) that operate on the data into a single unit or class, promoting data hiding and reducing complexity. • **Inheritance:** Grasping how new classes can be derived from existing ones, allowing for code reuse and the creation of hierarchical class structures. • **Polymorphism:** Understanding how different classes can be treated as instances of the same class through interfaces, allowing for flexibility in code through method overriding and overloading. • **Abstraction:** Learning to focus on essential qualities of an object while hiding unnecessary details, making complex systems easier to manage. **5- Developing Problem-Solving Skills:** • **Modeling Real-World Systems:** Teaching students to represent real-world entities as objects, helping to develop systems that are intuitive and maintainable. • **Design Patterns:** Introducing common design patterns that solve recurring problems in OOP, fostering best practices in software development. • **Code Reusability:** Emphasizing the importance of creating reusable, modular code that can be easily extended and maintained. **6- Improving Software Design and Architecture:** • **Software Design Principles:** Educating students on principles like SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) to create well-structured and robust code. • **Object-Oriented Analysis and Design (OOAD):** Training students to analyze and design software systems using OOP principles, focusing on creating scalable and maintainable architectures. **7- Enhancing Team Collaboration and Code Maintenance:** • **Version Control Integration:** Learning to use version control systems (e.g., Git) in the context of OOP projects to manage code changes collaboratively. • **Code Documentation and Comments:** Understanding the importance of documenting code, especially in large, object-oriented projects, to facilitate collaboration and maintenance. • **Testing and Debugging:** Gaining skills in writing unit tests for classes and objects, and learning debugging techniques specific to object-oriented codebases.
Module Learning Outcomes	**1. Knowledge and Understanding:** • **MLO 1:** Demonstrate a comprehensive understanding of the fundamental principles of Object-Oriented Programming, including concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction. • **MLO 2:** Understand and apply the principles of software design and architecture, including the use of design patterns and best practices in OOP. • **MLO 3:** Explain the benefits and limitations of the OOP paradigm in software development, including its impact on code reusability, maintainability, and scalability.

	2. Cognitive/Intellectual Skills: • **MLO 4:** Analyze real-world problems and design effective OOP solutions by modeling appropriate classes, objects, and relationships. • **MLO 5:** Critically evaluate and apply design patterns to solve common software design problems. • **MLO 6:** Assess the trade-offs between different object-oriented designs in terms of efficiency, complexity, and scalability. **3. Practical/Professional Skills:** • **MLO 7:** Develop and implement object-oriented software using a relevant programming language (e.g., Java, C++, Python) that adheres to industry standards and best practices. • **MLO 8:** Apply techniques for debugging, testing, and maintaining object-oriented code, including the use of unit tests and version control systems. • **MLO 9:** Work collaboratively in a team environment to design and develop a substantial object-oriented software project, demonstrating effective communication and project management skills. **4. Key Transferable Skills:** • **MLO 10:** Demonstrate problem-solving skills by breaking down complex problems into manageable components using OOP techniques. • **MLO 11:** Communicate technical information effectively, both verbally and in writing, through documentation, code comments, and presentations. • **MLO 12:** Adapt to new and emerging technologies in object-oriented programming, demonstrating lifelong learning and the ability to stay current with industry trends.
Indicative Contents	The indicative content of an Object-Oriented Programming (OOP) course includes an introduction to core concepts like classes, objects, inheritance, encapsulation, polymorphism, and abstraction, along with advanced topics such as composition vs. inheritance, design patterns, and SOLID principles. It also covers object-oriented analysis and design (OOAD), practical implementation in a chosen programming language, and testing/debugging techniques. Students will work on hands-on projects, including collaborative team development, integrating OOP with databases, and exploring modern frameworks and libraries. The course concludes with discussions on contemporary OOP languages, emerging trends, and the future direction of software development.

Learning and Teaching Strategies	
Strategies	• Lectures • Tutorials • Problem solving • Lab • Case study • Small project
Student Workload (SWL)	

Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	40	**Unstructured SWL (h/w)**	2.66
Total SWL (h/sem)	**100**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to inheritance
Weeks 2,3	Type of inheritance
Weeks 4,5,6	Constructor inheritance , friend function , virtual function
Week 7	Mid-term Exam
Weeks 8,9	UML Diagrams
Weeks 10 and 11	SOLID Principles
Weeks 12 and 13	Interfaces and abstract classes , Abstract Methods
Week 14	polymorphism
Week 15	template
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Weeks 1 and 2	Apply inheritance
Weeks 3 and 4	Methods and data member inheritance
Weeks 5,6 and 7	Constructor inheritance
Weeks 8 and 9	friend function, virtual function
Weeks 10,11 and 12	Interfaces and abstract classes , Abstract Methods
Weeks 13 and 14	Polymorphism examples.
Week 15	Template examples.

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	• Programming in C++ Frank Vahid and Roman Lysecky Available through the zyBooks website directly • A C++ compiler and/or IDE. There are many out there, but the only two that are officially supported: - CLion (on Windows and macOS) - Visual Studio (Windows only)	Yes
Recommended Texts	▪ Think Like a Programmer, An Introduction to Creative Problem Solving V. Anton Spraul ISBN: 978-1593274245 ▪ A good text editor, such as: Notepad++ (This is my personal favorite) Sublime Text Atom, or Vim, or anything else you might prefer	No
Websites	1-http://www.cplusplus.com/ 2-https://www.w3schools.com/cpp/	

Grading Scheme

Group	Grade	Marks %	Definition

Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Visual Programming**	**Module Delivery**	
Module Type	Core	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	COM-225		
ECTS Credits	4.00		
SWL (hr/sem)	100		
Module Level	2	**Semester of Delivery**	4
Administering Department		**College**	Type College Code
Module Leader	Muneer Maaroof Hassan	**e-mail**	Muneer_maaroof@uodiyala.edu.iq
Module Leader's Acad. Title	Asso. Prof.	**Module Leader's Qualification**	Msc.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

Prerequisite module	COM-211	**Semester**	3

Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	1. This course will provide a managerial perspective of information systems and what role they play in an organization. Student learn about the modern technologies and how organizations can use these technologies for their growth.
Module Learning Outcomes	CLO-1: Knowledge (C1) * Use the different elements of a visual programming language as building blocks to develop correct, coherent programs. CLO-2: Application (C3) * Program using the fundamental software development process, including design, coding, documentation, testing, and debugging. CLO-3: Analysis (C4) * Analyze problems, develop conceptual designs that solve those problems, and transform those designs to Visual Programs with VB.Net.
Indicative Contents	This course introduces computer programming using the Visual Programming language with object-oriented programming principles. Emphasis is on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger. Upon completion, students should be able to design, code, test and debug at a beginning level. This course has been approved to satisfy the Comprehensive Articulation Agreement for transferability as a pre-major and/or elective course requirement.

Learning and Teaching Strategies

Strategies	s.

Student Workload (SWL)

Structured SWL (h/sem)	63	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	37	**Unstructured SWL (h/w)**	2
Total SWL (h/sem)	**100**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

Week 1	Introduction to Scratch: Overview of the interface, sprites, blocks, and creating a basic project.
Week 2	Understanding Events: Event-driven programming and triggering actions using events.
Week 3	Loops and Iteration: Repeat blocks, forever loops, and nested loops.
Week 4	Conditional Statements: If-then and if-then-else blocks for decision-making.
Week 5	Variables: Creating and using variables to store and manipulate data.
Week 6	Working with Operators: Mathematical and logical operators for dynamic behavior.
Week 7	Creating Animations: Using costumes, backdrops, and timing for animations.
Week 8	Building Simple Games: Basics of game development, scoring, and timers.
Week 9	MIDTERM EXAM
Week 10	User Interaction: Sensing blocks for keyboard and mouse inputs.
Week 11	Broadcasting and Messaging: Communication between sprites using broadcast messages.
Week 12	Project Development: Designing and implementing a complete project (e.g., game or animation).
Week 13	Debugging and Optimization: Identifying and fixing errors in Scratch projects.
Week 14	Advanced Topics: Custom blocks (functions) and cloning.
Week 15	Final Project Preparation: Developing a comprehensive Scratch project.
Week 16	FINAL EXAM

Delivery Plan (Weekly Lab. Syllabus)

	Lab 1: Introduction to the Scratch interface and basic project creation.
Week 1	Lab 2: Event-driven programming with Scratch.
Week 2	Lab 3: Loops and iterations in Scratch projects.
Week 3	Lab 4: Using conditional statements in interactive projects.
Week 4	Lab 5: Creating and manipulating variables in Scratch.

Week 5	Lab 6: Adding animations with costumes and backdrops.
Week 6	Lab 7: Building a simple game with scoring and timers.
Week 7	Lab 8: Debugging and optimizing Scratch projects.
Week 8	Lab8: Implement the TRY-THROW-CATCH structure for exception handling

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	Mitchel Resnick, "Scratch Programming for Beginners: An Introduction to Creative Coding", Scratch Foundation. 650154-0.	Yes
Recommended Texts	- Derek Breen, "Scratch for Kids For Dummies", 2015, ISBN: 978-1119014874	No
Websites		

Grading Scheme

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

Course Description Form

Module Information

Module Title	**Computational mathematics**	**Module Delivery**

Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-112**		
ECTS Credits	**5**		
SWL (hr/sem)	**125**		

Module Level	1	**Semester of Delivery**	1
Administering Department	COM	**College**	cos
Module Leader	Khalid M.S. Al Zaidi	**e-mail**	dr.khaledmoh@uodiyala.edu.i
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

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

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	13. The principal objective of this course is to develop the analytic skills need t learn mathematics. 14. Studying basic mathematical concepts to solve problems. 15. To understand analyze systems in a mathematical manner. 16. This course deals with the basic concept of discrete mathematical. 17. This is the basic subject for most computer science subjects.
Module Learning Outcomes	Important: Write at least 6 Learning Outcomes, better to be equal to the number of study weeks. 1. Recognize the basic concepts in a discrete mathematical structure. 2. To understand the fundamental properties of sets. 3. Identify the basic sets operations.

	4. To study the sets types and counting principle. 5. Recognize the relations and functions to describe the relationship between the elements from two sets. 6. To learn several basic proof techniques. 7. Discuss the proof techniques to prove important results in set theory. 8. To studies properties of integers and use the proof techniques to prove some basic facts in number theory. 9. To understand the fundamental properties of graph. 10. To study how representation of functions by using graph. 11. Discuss the types of graphs and special graph. 12. Explain the Polish notation.
Indicative Contents	Indicative content includes the following. 1. Sets • The basic concepts • Set Operations • Finite sets, counting princple • Classes of sets • Partitions of set 2. Relations • Representation of relations • Properties of relations • Inverse relations • Composition of relations 3. Function • One-to-one, onto and invertible functions • Graph of a function • Composition of function 4. Graphs • Degree • Connectivity • Special graph • Matrices and graphs • Labeled graphs • Tree • Polish notation

Learning and Teaching Strategies	
Strategies	Discrete mathematics is foundational material for computer science: Many areas of computer science require the ability to work with concepts from discrete mathematics, specifically material from such areas as set theory, logic, graph theory, combinatorics, and probability theory.

	The main strategy that will be adopted in delivering the discrete mathematical structures module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. The module will include a combination of classes, and interactive tutorials.

Student Workload (SWL)

Structured SWL (h/sem)	52	**Structured SWL (h/w)**	3
Unstructured SWL (h/sem)	73	**Unstructured SWL (h/w)**	5
Total SWL (h/sem)	**125**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	20% (20)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction - The basic concepts
Week 2	Sets
Week 3	Set Operations
Week 4	Set Operations

Week 5	Finite sets, counting principle
Week 6	Cardinality
Week 7	Mid-term Exam
Week 8	Inverse relations
Week 9	Function
Week 10	Graph of a function
Week 11	Graphs
Week 12	Graphs
Week 13	Connectivity
Week 14	Special graph
Week 15	Polish notation
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

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

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	- Discrete mathematical structures for computer science by Bernard Kolman & Robert C. Busby	Yes
Recommended Texts	- Theory and problems of Discrete mathematics, by Seymour Lipschutz & Marc Lars Lipson, Schaum's Outline Series, third edition 2007. - Mathematical foundation of computer science, Y.N. Singh, 2005.	Yes

	- Discrete Mathematics and Its Applications, Seventh Edition, Kenneth H. Rosen, AT&T Laboratories, 2012.	
Websites	- http://www.math.uvic.ca/faculty/gmacgill/guide - http://en.wikibooks.org/wiki/Discrete_mathemati cs/Set_theory	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A -** Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Computer Skills (I)**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	**COM-113**		
ECTS Credits	**4**		
SWL (hr/sem)	**100**		

Module Level	1	**Semester of Delivery**	1
Administering Department	**COM**	**College**	COS
Module Leader	Name :muntadher Khamees mustafa	**e-mail**	E-mail alkarawis@gmail.com
Module Leader's Acad. Title	Assist. Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

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

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	**Students will explore and become more familiar with:** 9. The concepts of Information & Communication Technology (ICT) and its applications. 10. The operating systems such as Windows, Android, IOS, Linux, DOS and application software such as Word and Electronic Spreadsheets. 11. The continuous exchange of data between any two parts of the world. 12. Data visualization using different tools and applications. 13. The general programming concepts and related problem-solving strategies. 14. The design and development of applications using simple software/programming language. 15. The basic principles of information security.

Module Learning Outcomes	**Upon the completion of the course, students will be able to:** 1. The students will have the knowledge to understand the peripheral devices computer system and the different types of software and skills for managing visual elements, document sources, developing project reports, assignments etc. Which are mandatory at this level of academics. 2. The students will have the knowledge and skills to create presentations that include academic project presentations, seminars, professional-grade presentations, employee training manuals, instructional materials, and kiosk slideshows. 3. The students will have the knowledge of the importance of data analysis as a part of strategic growth, allowing students to forecast trends and required actions. 4. Students will have the knowledge of working independently as well as with a group to deliver effective and well- documented software solutions to all problems. 5. Students will have the knowledge of the importance Artificial Intelligence, as well as various applications of Internet of Things that are widely used in the field of computing technologies. 6. Students can apply skills by working and completing software-related activities such as MS Word, MS Excel, MS PowerPoint and Python 7. Mastering the skills taught throughout the course will improve the productivity and the way students are conducting and presenting their work. This course provides students with the basic knowledge and skills that allow them to use different kinds of computer applications.
Indicative Contents	**Indicative content includes the following.** 1. Introduction To Course 2. Computing Basics and Text Processing Essentials 3. Presentation Graphics Essentials 4. Data analysis and Dashboard Technique (5. Algorithm and Python Programming concepts 6. Advanced Computing Technologies 7. Self-Learning(Elective) 8. Others (Exam, PCA)

Learning and Teaching Strategies

Strategies	The class will "meet" the equivalent of two one-hour & fifteen minutes for lecture/discussion each week. The lectures and discussions will be a combination of sychronous and asynchronous discussions using WebCT. Students must have access to the Internet to facilitate demonstrating and using software. Many of the assignments should stress hands-on applications by the students. Each student will be expected to participate in all lectures. Class participation by all is expected.

Student Workload (SWL)

Structured SWL (h/sem)	66	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	34	**Unstructured SWL (h/w)**	2
Total SWL (h/sem)	**100**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	**Introduction**
Week 2	**Components of Computer System**
Week 3	**Concept of Hardware and Software**
Week 4	**Concept of computing, data and information**
Week 5	**Connecting keyboard, mouse, monitor and printer to CPU**
Week 6	**Basics of Operating System**
Week 7	**mid exam**
Week 8	**Basics of popular operating system (LINUX, WINDOWS)**
Week 9	**Operating System Simple Setting**
Week 10	**File and Directory Management**

Week 11	**Basic of Computer Networks**
Week 12	**Popular Web Browsing Softwares**
Week 13	**Basics of E-mail**
Week 14	**Data security**
Week 15	**Document collaboration**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Knowing computer
Week 2	Operating Computer using GUI Based Operating System
Week 3	network and Operating Computer using GUI Based Operating System
Week 4	Managing files and folders
Week 5	Basic networking
Week 6	Operating System Simple Setting
Week 7	Exam
Week 8	File and Directory Management
Week 9	Basic of Computer Networks
Week 10	Popular Web Browsing Softwares
Week 11	Basics of E-mail
Week 12	Image editing
Week 13	Simple computer programming concepts
Week 14	Simple computer programming concepts
Week 15	Data security

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Textbooks	Computer Skills	Yes
Essential References Materials	LMS Black Board.	No
Electronic Materials	Electronic Materials	

Grading Scheme

Group	**Grade**	**Marks %**	**Definition**

Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Digital Logic**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-114**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level		**Semester of Delivery**	1
Administering Department	COM	**College**	cos
Module Leader	Yahiea M.H. Al Naiemy	**e-mail**	Yahiea.alnaiemy@uodiyala.edu.iq
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

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

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	This is core course of Computer Science Department and that presents basic tools for the design of digital circuits. It serves as a building block in many disciplines that utilize data of digital nature like digital control, data communication, digital computers etc. The goal of this course is to; 1. Perform arithmetic operations in many number systems. 2. Manipulate Boolean algebraic structures. 3. Simplify the Boolean expressions using Karnaugh Map (K-MAP). 4. Implement the Boolean Functions using NAND and NOR gates. 5. Analyze and design various combinational logic circuits such as (Binary Adder and Substructor, Multiplexer, Decoder, Programmable Logic Array (PLA)). 6. Understand the basic functions of Flip-Flops. 7. Understand the importance of state diagram representation of sequential circuits. 8. Analyze and design clocked sequential circuits such as Counters and Registers.
Module Learning Outcomes	This course makes significant contributions to the following program outcomes: 1. An ability to apply knowledge of mathematics, science, and engineering, 2. An ability to design and conduct experiments, as well as to analyze and interpret data, 3. An ability to design a system, component , or process to meet desired needs within realistic constraints 4. An ability to identify, formulate, and solve Science and engineering problems, 5. An ability to use the techniques, skills, and modern engineering tools necessary for Computer and Engineering practice.
Indicative Contents	1. Digital Systems and Binary Numbers • Digital Systems, • Binary Numbers, • Number-base Conversions, • Octal and Hexadecimal Numbers, • Complements, • Signed Binary Numbers, • Binary Codes, Binary Storage and Registers, • Binary Logic. 2. Boolean Algebra and Logic Gates: • Introduction, • Basic Definitions,

	• Axiomatic Definition of Boolean Algebra, • Basic Theorems and Properties of Boolean Algebra, • Boolean Functions, • Canonical and Standard Forms, • Other Logic Operations, • Digital Logic Gates, • Integrated Circuits. 3. Gate level Minimization: • The Map Method, • Two-variable map, • Three-variable map, • Four-Variable Map, • Five-variable Map, • Product of Sums Simplification, • Don't-care Conditions, • NAND and NOR Implementation, • Other Two-Level Implementations, • Exclusive-OR and NOR Function. 4. Combinational Logic: • Introduction, • Combination Circuits, • Analysis Procedure, • Design Procedure, • Binary Adder-Subtractor, • Decimal Adder, • Binary Multiplier, • Magnitude Comparator, • Decoders, • Encoders, • Multiplexers and DeMultiplexers. 5. Synchronous Sequential Logic: • Introduction, • Sequential Circuits, • Storage Element: • Latches, • Storage Element: Flip-Flops, • Analysis of Clocked Sequential Circuits, • State Reduction and Assignment, • Design Procedure. • Synchronous Counter Design • Design Sequential Logic: Shift Registers.

Learning and Teaching Strategies

Strategies	The main strategy that will be adopted in delivering the logic design module is to engage students actively in practical exercises to enhance their understanding and develop their critical thinking skills. The module will include a combination of classes, interactive tutorials, and hands-on experiments focused on sampling activities that capture students' interest. Through interactive classes, students will be introduced to the theoretical foundations of logic design, including explain digital system concept. express analog to digital conversion, use binary number system, realize conversion between various number systems, design fundamental digital systems, recognize logic gates, apply

Boolean algebra, employ Karnaugh map for digital system optimization, develop combinational logic circuits such as adder, subtractor, encoder, decoder, multiplexer and demultiplexer. And recognize types of Flip-flops, design sequential logic circuits. Analyze fundamental digital systems, calculate input - output relationship in digital systems, and recognize state diagrams and tables, analyses sequential logic circuits.

To reinforce learning and encourage active participation, interactive tutorials will be conducted. These tutorials will involve hands-on exercises where students will work with real-world design and apply different design techniques. This practical approach will help students grasp the practical implications of the theoretical concepts discussed in the classes.

In addition to tutorials, simple experiments will be introduced to provide students with opportunities to explore various sampling activities. These experiments will focus on real-life scenarios and problems related to digital design . Students will be encouraged to think critically, analyze the results, and propose solutions based on their understanding of the concepts learned.

The module will also emphasize the importance of collaboration and teamwork. Students will be encouraged to work together on projects and assignments, fostering a collaborative learning environment where they can exchange ideas and learn from each other's perspectives.

Overall, the module's delivery approach aims to actively engage students, refine their critical thinking skills, and provide them with practical experiences in digital design . By combining theoretical knowledge with hands-on activities, students will develop a deeper understanding of logic design concepts and their applications in various fields.

Student Workload (SWL)

Structured SWL (h/sem)	95	**Structured SWL (h/w)**	6
Unstructured SWL (h/sem)	55	**Unstructured SWL (h/w)**	3
Total SWL (h/sem)	**150**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 9	LO #1, #2 and #10, #11

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction –Logic Design and its application
Week 2	Arithmetic Operations
Week 3	Logic Gates
Week 4	Simplification and Boolean Functions
Week 5	Logic Operations
Week 6	Combinational and Sequential Circuit Analysis Design
Week 7	Mid-term Exam
Week 8	Digital Circuit Design Optimization Methods
Week 9	Binary Adder and Substructor
Week 10	Multiplexer, Decoder, PLA
Week 11	Types of RAMs ,ROMs
Week 12	Programmable Logic Arrays
Week 13	Flip-Flops
Week 14	Counters
Week 15	Registers
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Lab 1: Digital binary representation
Week 2	Lab 2: Basic Gates Electronic WorkBench
Week 3	Lab 3: Two input SOP and Two input POS
Week 4	R-S flip-flop. J - K flip-flop, T Flip-Flop, D Flip-Flop
Week 5	Lab 5: - Implementation and verification of decoder, de-multiplexer and encoder, using logic gates.
Week 6	Lab 6: Implementation of 4x1 multiplexer, using logic gates.
Week 7	Lab 7: Implementation of 4-bit parallel adder, using 7483 IC

Week 8	Lab 8: Design, and verify the 4-bit synchronous counter.

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	• M. Morris Mano, "Computer System Architecture", 1998.	Yes
Recommended Texts	• Donald D. Givone (2002), Digital Principles and Design, Tata McGraw Hill. • Roth (2004), Fundamentals of Logic Design, 5th Edition, Thomson.	No
Websites	https://www.circuitlab.com/editor/#?id=7pq5wm&from=homepage https://circuitverse.org/simulator	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Introduction to Programming**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	COM-111		
ECTS Credits	8		
SWL (hr/sem)	200		

Module Level		**Semester of Delivery**	1
Administering Department	COM	**College**	COS
Module Leader	Taha Mohammed Hasan	**e-mail**	dr.tahamh@uodiyala.edu.iq
Module Leader's Acad. Title	Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

Prerequisite module	None	**Semester**	
Co-requisites module	Programming Fundamentals	**Semester**	2

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	The educational objectives of this course are 1- To Focus Fundamentals of Computers and Peripherals 2- To Introduce programming language and aware the students about programming paradigm 3- To Focus Concept and Methodology of Programming

	4- Brief the students regarding Object Oriented Programming Features 5- To give clear idea of different strategy of basic programming with C like Looping, Decision Making, Array, Structure, Function, Pointer, etc. to solve real life problems.
Module Learning Outcomes	1- On successful completion of the course, the student will be having the basic knowledge of programming paradigm, fundamentals of compute and peripherals and thus being prepared with the programming spectrum in depth as desired. 2- Student will be able to effectively solve any real-life problem and lead the exploration of new application and techniques for their use.

Indicative Contents	• Introduction to Computer Programming
	• Basics of C++ language
	• Problem Solving and Algorithm Design
	• Pseudo-codes and Flow charts
	• Arithmetic Operators and Variables
	• Exploring input and output statements
	• Control Structure (Selection and iterative)
	  ○ Functions
	• Primary data structure of Arrays and its multi – dimensional behavior.
	  ○ Concepts of Pointers
	• Introductory knowledge of Structures

Learning and Teaching Strategies	
Strategies	• Lectures • Tutorials • Problem solving • Lab • Case study • Small project

Student Workload (SWL)			
Structured SWL (h/sem)	109	**Structured SWL (h/w)**	7
Unstructured SWL (h/sem)	91	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	**200**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction – History of programming languages. Low-level languages and High-level languages.
Week 2	Problem solving - Flowcharts and pseudocode algorithms.
Weeks 3,4,5 and 6	Introduction to C/C++ programming language: • History of C/C++ • C++ standard Library. • C++ Environment. • General structures of C/C++ • programming language. • Data types. • Variables declaration/definition. • Directives. • Inputs and outputs. Simple programming
Week 7	Mid-term Exam
Week 8	Arithmetic and operators: Arithmetic operators. Operator's precedence. Equality and relational operators. Sequences.

Weeks 9 and 10	Control Structures: • Selection and Decisions: • if • if…else. • nested if • switch
Weeks 11,12 and 13	Control Structures: • Iteration: • for • while • do while
Weeks 14 and 15	Array: • Array definition (one-dimensional array). • operations on array (add, subtraction, multiplication and invers of array).
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Problem solving and Algorithms
Week 2	Introduction to C/C++ Integrated development Environments (IDE).
Weeks 3 and 4	Introduction to C/C++ programming. Writing simple programs that involve using input/output statements. identify and fix common syntax errors.
Weeks 5 and 6	Data type, Operators, and Expressions
Weeks 7,8,9 and 10	Control structure writing program using if, if..else, switch, for, while &do…while control structure
Weeks 11,12 and 13	Array (one-dimensional array). Operations on array

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	• Deitel & Deitel, 2017, "C++ How to Program", Tenth Edition, Pearson Education. • D. S. Malik, 2018, "C++ programming from Problem Analysis to Program Design", Eighth Edition, Cengage Learning.	Yes
Recommended Texts	• Stanley B Lippman; Josée Lajoie; Barbara E, 2013, "C++ PRIMER",5th Edition, Addison Wesley	No
Websites	1-http://www.cplusplus.com/ 2-https://www.w3schools.com/cpp/	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A -** Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information					
Module Title	**Computer Organization**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-123**				
ECTS Credits	**6**				
SWL (hr/sem)	**150**				
Module Level	1		**Semester of Delivery**	2	
Administering Department	Comp		**College**	COS	
Module Leader	Bashar Talib AL-Nuaimi		**e-mail**	Alnuaimi _**bashar@uodiyala.edu.iq**	
Module Leader's Acad. Title	Ass. Prof.		**Module Leader's Qualification**	Ph.D.	
Module Tutor	Name (if available)		**e-mail**	E-mail	
Peer Reviewer Name	Name		**e-mail**	E-mail	
Scientific Committee Approval Date	01/06/2023		**Version Number**	1.0	

Relation with other Modules			
Prerequisite module	COM-113	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	1- To impart basic concepts of computer architecture and organization, 2- To explain key skills of constructing cost-effective computer systems. 3- To familiarize the basic CPU organization. 4- To help students in understanding various memory devices. 5- To facilitate students in learning IO communication
Module Learning Outcomes	1- Identify various components of computer and their interconnection 2- Identify basic components and design of the CPU: the ALU and control unit. 3- Compare and select various Memory devices as per requirement. 4- Compare various types of IO mapping techniques 5- Critique the performance issues of cache memory and virtual memory
Indicative Contents	Indicative content includes the following. The module focuses on computer system hardware and how it is used to facilitate the execution of software, including topics such as data representation, digital logic, assembler, memory systems, I/O and microprocessor architecture. The aim of the module is to provide students with an understanding of the functional components of a computer system and how they are organized to facilitate execution. Student will also gain a range of practical skills in the application and construction of computer components that are capable of interfacing with microprocessors. This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ. Data Representation: Analog and digital data; number systems including number representations and conversions; binary variables including logic expressions and simplifications of binary variables; binary arithmetic, signed numbers; arithmetic structures, multiple precision arithmetic, Microprocessor architecture; instruction cycles, register transfer language (RTL); instruction set and addressing modes; assembly and machine code programming. Input/Output Mechanisms: Parallel I/O; memory-mapped and programmable I/O devices, eg VIA; I/O scheduling including programmed I/O, interrupt driven I/O, direct memory access (DMA); analog I/O. Memory systems: Motivation for memory hierarchy; devices including magnetic tape, floppy disks, disk formats, hard disks, optical data storage; elementary error detection and correction. Processor Architecture: Example combination of ALU, registers, and CPU; instruction set construction; control units including hardwired and microprogrammed control units.

Learning and Teaching Strategies

Strategies	Type something like: The main strategy that will be adopted in delivering this module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. This will be achieved through classes, interactive tutorials and by considering types of simple experiments involving some sampling activities that are interesting to the students.

Student Workload (SWL)

Structured SWL (h/sem)	78	**Structured SWL (h/w)**	5
Unstructured SWL (h/sem)	47	**Unstructured SWL (h/w)**	3
Total SWL (h/sem)	**150**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to computer systems Main parts of computer system, organization and architecture
Week 2	Von Neumann architecture
Week 3	Computer generations
Week 4	Data presentation
Week 5	Introduction to Computer components (motherboard components in details (i.e cpu and memory details))
Week 6	INPUT OUTPUT SYSTEM: I/O Devices, Video Color Models
Week 7	CPU, CPU basic organization, Control units
Week 8	MEMORY SYSTEM, Memory hierarchy,
Week 9	Mid-term Exam
Week 10	Cache memory, Primary memory, Secondary memory, memory addressing,
Week 11	Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost,
Week 12	Introduction to Secondary Storage
Week 13	Hard disk and magnetic drive
Week 14	Computer s/w
Week 15	Machine language, high level language
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Lab 1 Introduction to computer systems
Week 2	Lab 2: Overview of computer hardware and software
Week 3	Lab 3: Introduction to windows
Week 4	Lab 4: Shortcuts and Start menu.
Week 5	Lab 5: Open WordPad from the shortcut on the desktop. Open WordPad from the Start menu
Week 6	Lab 6: Introduction to Notepad and WordPad. Typing a letter. Opening a file and saving a file
Week 7	Lab 7: Opening a text document from a CD. Correcting a text document
Week 8	Lab 8: Mid-term Exam
Week 9	Lab 9: Working with multiple windows.

Week 10	Lab 10: Creating a folder and saving a document to that folder. Deleting a file. The Recycle Bin.
Week 11	Lab 11: formatting windows
Week 12	Lab 12: Desktop Customizations
Week 13	Lab 13: Control Panel
Week 14	Lab 14: Internet and Computer Networks
Week 15	Preparatory week before the final Exam
Week 16	

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	Computer Architecture: A Quantitative Approach (5th edition) by J.L. Hennessy and D.A. Patterson (Morgan Kauffmann Publishers)	Yes
Recommended Texts	Computer System Architecture by M. Morris Mano (Pearson Publication)	No
Websites	https://www.youtube.com/watch?v=Ol8D69VKX2k&t=147s	

Grading Scheme

Group	Grade	Marks %	Definition
Success Group (50 - 100)	**A -** Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information					
Module Title	**Discrete Structures**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**COM-122**				
ECTS Credits	**5**				
SWL (hr/sem)	**125**				
Module Level	1		**Semester of Delivery**	2	
Administering Department	com		**College**	cos	
Module Leader	Khalid M.S. Al Zaidi		**e-mail**	dr.khaledmoh@uodiyala.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	Ph.D.	
Module Tutor	Name (if available)		**e-mail**	E-mail	
Peer Reviewer Name	Name		**e-mail**	E-mail	
Scientific Committee Approval Date	01/06/2023		**Version Number**	1.0	

Relation with other Modules			
Prerequisite module	COM-112	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	18. The principal objective of this course is to develop the analytic skills need t

	learn mathematics. 19. Studying basic mathematical concepts to solve problems. 20. To understand analyze systems in a mathematical manner. 21. This course deals with the basic concept of discrete mathematical. 22. This is the basic subject for most computer science subjects.
Module Learning Outcomes	Important: Write at least 6 Learning Outcomes, better to be equal to the number of study weeks. 13. Recognize the basic concepts in a discrete mathematical structure. 14. To understand the fundamental properties of modeling computation. 15. Identify the finite state machines. 16. To study an optimistic approach principle. 17. Recognize the finite automata. 18. To studies properties deterministic finite state automata. 19. Recognize the propositions and truth values 20. To understand the logical connectives 21. To study tautologies and contradictions principle 22. To studies properties of logical equivalence 23. Recognize the algebra of propositions 24. Identify of the mathematical Induction 25. To studies methods of proofs by mathematical induction 26. To studies properties of matrices, types, and an operations on matrices
Indicative Contents	Indicative content includes the following. 1. Modeling Computation • Finite state machines • An Optimistic Approach • Finite automata • Deterministic Finite State Automata 2. Logic and Proofs • Propositions and Truth Values • Logical connectives • Tautologies and Contradictions • Logical Equivalence • The Algebra of propositions • Mathematical Induction 3. Matrices • Types of Matrices • Operations on Matrices

Learning and Teaching Strategies	
Strategies	Discrete mathematics is foundational material for computer science: Many areas of computer science require the ability to work with concepts from discrete

	mathematics, specifically material from such areas as set theory, logic, graph theory, combinatorics, and probability theory. The main strategy that will be adopted in delivering the discrete mathematical structures module is to encourage students' participation in the exercises, while at the same time refining and expanding their critical thinking skills. The module will include a combination of classes, and interactive tutorials.

Student Workload (SWL)

Structured SWL (h/sem)	64	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	61	**Unstructured SWL (h/w)**	4
Total SWL (h/sem)	**125**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	20% (20)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction - Modeling Computation
Week 2	Finite state machines
Week 3	An Optimistic Approach

Week 4	Finite automata
Week 5	Deterministic Finite State Automata
Week 6	Logic and Proofs
Week 7	Mid-term Exam
Week 8	Propositions and Truth Values
Week 9	Tautologies and Contradictions
Week 10	Logical Equivalence
Week 11	The Algebra of propositions
Week 12	Mathematical Induction
Week 13	Matrices
Week 14	Types of Matrices
Week 15	Operations on Matrices
Week 16	**Preparatory week before the final Exam**

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

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	- Discrete mathematical structures for computer science by Bernard Kolman & Robert C. Busby	Yes
Recommended Texts	- Theory and problems of Discrete mathematics, by Seymour Lipschutz & Marc Lars Lipson, Schaum's Outline Series, third edition 2007.	Yes

	- Mathematical foundation of computer science, Y.N. Singh, 2005. - Discrete Mathematics and Its Applications, Seventh Edition, Kenneth H. Rosen, AT&T Laboratories, 2012.	
Websites	- http://www.math.uvic.ca/faculty/gmacgill/guide	

Grading Scheme			
Group	**Grade**	**Marks %**	**Definition**
Success Group (50 - 100)	**A -** Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information		
Module Title	**Integration Methods**	**Module Delivery**
Module Type	**BASIC**	☒ **Theory** ☒ **Lecture** ☐ **Lab** ☐ **Tutorial**
Module Code	**COS-102**	
ECTS Credits	**5**	

SWL (hr/sem)	**125**	☐ **Practical** ☐ **Seminar**	
Module Level	1	**Semester of Delivery**	2
Administering Department	COM	**College**	COS
Module Leader	Hassan Kadhom I*brahim*	**e-mail**	hassan.kadhom.ibrahim@uodiyala.edu.iq
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	04/06/2023	**Version Number**	1.0

Relation with other Modules

Prerequisite module	COS -101	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	23. Differentiate algebraic and trigonometric functions using key 24. Find the tangent line to a given graph at a given point
Module Learning Outcomes	Upon successful completion of this course, students will: 1. Use differentiation rules to differentiate algebraic and transcendental functions. 2. Identify appropriate calculus concepts and techniques to provide mathematical models of real-world situations and determine solutions to applied problems. 3. Evaluate definite integrals using the Fundamental Theorem of Calculus. 4. Demonstrate an understanding of the relationship between derivatives and integrals using the Fundamental Theorem of Calculus.
Indicative Contents	• Applications of Differentiation Maximum and minimum, rates of change, differentials, linear approximations, increasing and decreasing functions, curve sketching. • Integration Integrals, techniques of integration, applications of integration. • Ordinary Differential Equations First order equations, second order linear equations with constant coefficients.

Learning and Teaching Strategies

Strategies	Develop an effective and time-efficient **homework/study strategy** for, not only your calculus class, but other classes as well. This will help you become a more confident, successful, and well-rounded student. It will lead to a healthier balance between work time and leisure time. Spend at least **two to four hours** on each homework assignment. This affords you extra time to work on challenging homework problems and helps you organize your thoughts, questions, and ideas. The more time you spend on homework, the more likely you are to articulate clear, concise questions to your classmates and teachers. The more time you spend on homework, the less time you will spend on frantic, last-minute preparation for exams. Definitions, formulas, and theorems that are introduced in class or needed to complete homework assignments should be **memorized immediately** . Postponing this until it's needed for the exam will impede your work speed on homework assignments and interfere with clearer and deeper understanding of calculus. Spend time working on calculus **every day** . Doing some calculus every day makes you more familiar with concepts, definitions, and theorems. This familiarity will make calculus get easier and easier one day at a time. Find at least **one or two other students** from your calculus class with whom you can regularly do homework and prepare for exams. Your classmates are perhaps the least used and arguably your best resource. An efficient and effective study group will streamline homework and study time, reduce the need for attendance at office hours, and greatly improve your written and spoken communication. The best time to use your classmates as study/homework partners is after you have made an honest effort on your own to solve the problems using your own wits, knowledge, and experience. When you encounter an unsolvable problem, don't give up too soon on it. Being stumped is an opportunity for mathematical growth and insight, even if you never solve the problem on your own. If you seek help prematurely, you will never know if you could have solved a tough problem without outside assistance. Begin preparing/outlining for exams at least **five class days** before the exam. Outlining the topics, definitions, theorems, equations, etc. that you need to know for the exam will help you focus on those areas where you are least prepared. Preparing early for the exam will build your self-confidence and reduce anxiety on the day of the exam. It's also an insurance policy against time lost to illness, unexpected family visits, and last-minute assignments in other classes. Generally speaking, pulling all-nighters and doing last-minute cramming for exams is a recipe for eventual academic disaster. Prepare for exams by working on **new problems** . Good sources for these problems are unassigned problems from your textbook, review exercises and practice exams at the end of each chapter, old hour exams, or old final exams. Studying exclusively from those problems which you have already been assigned and worked on may not be effective exam preparation. Problems for each topic are generally in the same section of the book, so knowing how to do a problem because you know what section of the book it is in could give you a false sense of security. Working on new randomly mixed problems more closely simulates an exam situation, and requires that you both categorize the problem and then solve it. Use **all** resources of assistance and information which are available to you. These include classnotes, homework solutions, office hours with your professor or teaching assistants, and problem sessions with your classmates. Do not rely exclusively on just one or two of these resources. Using all of them will help you develop a broader, more natural base of knowledge and understanding. Expect your exams to be **challenging** . If they are challenging, you will be prepared. If they are not challenging, you can expect to have an easy time getting a very high score !

Student Workload (SWL)

Structured SWL (h/sem)	64	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	61	**Unstructured SWL (h/w)**	4
Total SWL (h/sem)	**125**		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	10% (10)	5 and 10	LO #1, #2 and #10, #11
	Assignments	2	10% (10)	2 and 12	LO #3, #4 and #6, #7
	Projects / Lab.				
	Report	1	10% (10)	13	LO #5, #8 and #10
Summative assessment	**Midterm Exam**	2hr	20% (20)	7	LO #1 - #7
	Final Exam	3hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Differentiation: Introducing the Derivative. The Derivative as a Function. Rules of Differentiation
Week 2	Differentiation: The Product and Quotient Rules. Derivatives of Trigonometric Functions. Derivatives as Rates of Change.
Week 3	Differentiation: The Chain Rule, Implicit Differentiation
Week 4	Differentiation: Derivatives of Inverse Trigonometric Functions. Related Rates.
Week 5	Applications of Derivatives: Maxima and Minima. Mean Value Theorem. What Derivatives Tell Us
Week 6	Applications of Derivatives: Graphing Functions. Optimization Problems
Week 7	Applications of Derivative: Linear Approximation and Differentials. L'Hôpital's Rule
Week 8	Integration: Approximating Areas under Curves. Definite Integrals. Fundamental Theorem of Calculus.
Week 9	Integration: Working with Integrals. Substitution Rule. Part1
Week 10	Integration: Working with Integrals. Substitution Rule. Part2

Week 11	Integration: Working with Integrals. Substitution Rule. Part 3
Week 12	Integration: Regions Between Curves. Volume by Slicing. Volume by Shells. Part1
Week 13	Integration: Regions Between Curves. Volume by Slicing. Volume by Shells. Part2
Week 14	Integration: Length of Curves, Surface Area, Logarithmic and Exponential Functions.
Week 15	Review
Week 16	**Preparatory week before the final Exam**

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	Edwards, C.H. and Penney, D.E. Elementary Differential Equations. Prentice-Hall. (latest ed.).	Yes
Recommended Texts	Thomas, G. and Finney, R. Calculus and Analytic Geometry. Addison-Wesley. (latest ed.). Adams, R. Single Variable Calculus. Pearson Education. (latest ed.).	Yes
Websites		

Grading Scheme

Group	**Grade**	**Marks %**	**Definition**
Success Group (50 - 100)	**A** - Excellent	90 - 100	Outstanding Performance
	B - Very Good	80 - 89	Above average with some errors
	C - Good	70 - 79	Sound work with notable errors
	D - Satisfactory	60 - 69	Fair but with major shortcomings
	E - Sufficient	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	(45-49)	More work required but credit awarded
	F – Fail	(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

Module Information

Module Title	**Programming Fundamentals**	**Module Delivery**	
Module Type	**Core**	☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☒ **Seminar**	
Module Code	COM-121		
ECTS Credits	8		
SWL (hr/sem)	200		

Module Level	1	**Semester of Delivery**	2
Administering Department	com	**College**	cos
Module Leader	Taha Mohammed Hasan	**e-mail**	dr.tahamh@uodiyala.edu.iq
Module Leader's Acad. Title	Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Name (if available)	**e-mail**	E-mail
Peer Reviewer Name	Name	**e-mail**	E-mail
Scientific Committee Approval Date	01/06/2023	**Version Number**	1.0

Relation with other Modules

Prerequisite module	**Introduction to Programming**	**Semester**	1
Co-requisites module	None	**Semester**	

Module Aims, Learning Outcomes and Indicative Contents

Module Objectives	The educational objectives of this course are 6- Demonstrate a thorough understanding of modular programming by designing programs that require the use of programmer-defined functions. 7- Demonstrate a thorough understanding of arrays by designing and implementing programs that search and sort arrays. 8- Demonstrate a thorough understanding of the object-oriented programming concepts of encapsulation, data abstraction and

	composition by designing and implementing classes including the use of overloaded functions and constructors. 9- Demonstrate a thorough understanding of the concept of pointers and dynamic memory allocation by designing and implementing programs using pointers and dynamic memory allocation. 10- Demonstrate a thorough understanding of the implementation of programmer-defined functions and classes by writing code, performing unit testing and debugging of multiple complex programs. 11- Demonstrate good documentation style in all of the programs written in this course. 12- Demonstrate proficiency in implementing data validation code, performing unit testing, and developing test plans while implementing robust solutions to the assignments in this course. 13- Demonstrate a thorough understanding of stream input/output for both console and files. 14- Demonstrate an understanding of the differences between C and C++ in the areas of strings, pass by reference/passing pointers, and structs by designing and implementing programs that use C strings, C++ strings, C language structs and classes.
Module Learning Outcomes	Students will be exposed to the following concepts and/or skills at an introductory concepts level: 1- The analysis and design of programs based on requirements and performance considerations. 2- evaluation of various possible technical solutions. 3- object-oriented design consideration. 4- system integration. 5- program documentation. 6- program debugging procedures. 7- developing program testing plans. 8- consideration of program operating environment. 9- use of reusable software.
Indicative Contents	Introduction to the C++ programming language and its subset, Program structure, blocks, storage types, console and file I/O, functions, arrays, strings, pointers, call-by-reference, call-by-value, and dynamic memory allocation. The concept and use of classes will be covered in some detail. Differences between C, C++. Some new features in C++ will be introduced.

Learning and Teaching Strategies

Strategies	• Lectures • Tutorials • Problem solving • Lab • Case study • Small project

Student Workload (SWL)

Structured SWL (h/sem)	109	**Structured SWL (h/w)**	7
Unstructured SWL (h/sem)	91	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	**200**		

Module Evaluation

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

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to computers & programming.
Weeks 2,3	Array: • Array definition (Two-dimensional array). • operations on Two- dimensional array (add, subtraction, multiplication and invers of array).
Weeks 4,5,6	Functions
Week 7	Mid-term Exam
Weeks 8,9	Introduction to Classes & Objects
Weeks 10 and 11	Searching, Sorting, Algorithm Performance Analysis
Weeks 12 and 13	Pointers, dynamic memory allocation
Week 14	More about Classes and OOP

Week 15	Recursion
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Weeks 1 and 2	Array (Two-dimensional array). Operations on tow-dimensional array
Weeks 3 and 4	Decisions, Loops, Functions
Weeks 5,6 and 7	Classes
Weeks 8 and 9	Searching/Sorting
Weeks 10,11 and 12	Searching/Sorting
Weeks 13 and 14	Dynamic Arrays, Pointers
Week 15	Recursion

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	• Programming in C++ Frank Vahid and Roman Lysecky Available through the zyBooks website directly • A C++ compiler and/or IDE. There are many out there, but the only two that are officially supported: - CLion (on Windows and macOS) - Visual Studio (Windows only)	Yes
Recommended Texts	▪ Think Like a Programmer, An Introduction to Creative Problem Solving V. Anton Spraul ISBN: 978-1593274245 ▪ A good text editor, such as: Notepad++ (This is my personal favorite) Sublime Text Atom, or Vim, or anything else you might prefer	No
Websites	1-http://www.cplusplus.com/ 2-https://www.w3schools.com/cpp/	

Grading Scheme

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