MODULE DESCRIPTION FORM

Module Information

Module Title	**Computer Networks**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2101**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	1
Administering Department	Information Technology	**College**	College of Science
Module Leader	**Karar Sadiq mojsin**	**e-mail**	karar.sadeq@uokerbala.edu.iq
Module Leader's Acad. Title	Asst. Lecturer	**Module Leader's Qualification**	MS.c
Module Tutor	Ali abdulhussein ibrahim	**e-mail**	ali.abdulhussein19@uowa.edu.iq
Peer Reviewer Name	Asst . Lect Nabeel Sadeq	**e-mail**	nabeel.alshreefy@uowa.edu.iq
Scientific Committee Approval Date	2024-09-17	**Version Number**	V01

Relation with other Modules

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

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د ضياء حسين نوفل
٢٠٢٥/٢٠٢٤

أ.م.د ضياء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	The "Computer Networks" module aims to provide IT undergraduate students with a solid foundation in computer networks. The module starts with an introduction to networks and progressively delves into the application and transport layers. Through a combination of theoretical knowledge and practical applications, the module aims to enable students to comprehend the principles, protocols, and functionalities of computer networks. By the end of the module, students should be capable of analyzing network requirements, designing appropriate solutions, implementing network services, and diagnosing common issues at the application and transport layers. Furthermore, the module aims to foster critical thinking, problem-solving skills, and an understanding of best practices for securing computer networks. Ultimately, the module seeks to prepare students for professional roles in network administration, network engineering, and related fields by equipping them with the necessary knowledge and skills in computer networks.
Module Learning Outcomes	Understand the fundamental concepts and principles of computer networks. Analyze and explain the functionalities and protocols of the application and transport layers. Evaluate network requirements and design appropriate solutions for different scenarios. Implement and configure network services and protocols at the application and transport layers. Diagnose and troubleshoot common network issues at the application and transport layers. Apply best practices for securing computer networks at the application and transport layers.
Indicative Contents	Introduction to Networks Overview of computer networks and their importance in modern IT infrastructure. Network topologies, protocols, and standards. Network architectures: client-server, peer-to-peer, hybrid. Network components: routers, switches, hubs, and cables. Application Layer Overview of the application layer and its role in network communication. Common application layer protocols: HTTP, FTP, DNS, SMTP. Application layer services: email, web browsing, file transfer. Socket programming and network application development. Transport Layer Overview of the transport layer and its role in reliable data delivery. Transport layer protocols: TCP and UDP. Flow control, congestion control, and error detection techniques. Quality of Service (QoS) considerations at the transport layer.

Learning and Teaching Strategies	
Strategies	Lectures: In-class lectures will be delivered to introduce and explain key concepts, theories, and principles related to computer networks. Lectures will include real-world examples and case studies to enhance understanding. Practical Sessions: Practical sessions will provide hands-on experience in configuring and managing computer networks. Students will have the opportunity to work with networking tools, simulate network scenarios, and troubleshoot network issues. Group Discussions: Group discussions will encourage students to critically analyze and discuss networking concepts, protocols, and design principles. This will foster collaborative learning and the exchange of ideas among peers. Case Studies and Projects: Students will be assigned case studies and projects that require them to apply their knowledge and skills to real-world network scenarios. This will help them develop problem-solving abilities and reinforce their understanding of network concepts. Independent Study: Students will be expected to engage in independent study to further explore and deepen their understanding of the module content. This may involve reading recommended textbooks, researching additional resources, and completing assigned exercises. Assessments: Assessments will include individual and group assignments, practical exercises, quizzes, and examinations. These assessments will evaluate students' understanding of concepts, ability to apply knowledge, and skills in network analysis and troubleshooting.

Student Workload (SWL)			
Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	8	10% (8)	2,4,6,8,10	1,2,3,4
	Home Work	5	10% (7)	Continues	3,5,7,9,11
	Projects	1	10% (5)	9	1,2,3,4,5,6,7
	Lab	5	10%(15)	Continues	1,2,3,4,5,6,7
	Onsite Assignmnets	5	10% (5)	1,2,3,4,5,6,8,9, 10,11,12,13,14, 15	3,5,7,9,11
Summative assessment	Midterm Exam	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to Networks
Week 2	Network Core: Packet and Circuit Switching
Week 3	Delay, Loss, Throughput in Networks
Week 4	Protocol Layers and Service Model
Week 5	Principles of Network Applications
Week 6	Web and HTTP FTP
Week 7	Electronic Mail: SMTP, POP3, IMAP
Week 8	DNS and P2P
Week 9	Transport Layer: Services
Week 10	Multiplexing and Demultiplexing
Week 11	Reliable Data Transfer RDT
Week 12	Connectionless Transport Protocol: UDP
Week 13	Connection-oriented transport: TCP
Week 14	TCP Congestion Control

Week 15	Flow Control
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Introduction to Network Components and Configurations
Week 2	Analyzing Network Topologies
Week 3	Configuring and Testing Network Protocols
Week 4	Socket Programming
Week 5	HTTP and FTP
Week 6	Flow Control and Congestion Control
Week 7	Quality of Service (QoS) Configuration
Week 8	Network Security and Firewalls
Week 9	Virtual Private Networks (VPNs)
Week 10	Network Monitoring and Troubleshooting
Week 11	SMTP, IMAP and POP3
Week 12	Network Address Translation (NAT)
Week 13	DNS Configuration and Domain Setup
Week 14	Network Virtualization
Week 15	Network Performance Testing and Optimization

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	L. L. Peterson and B. S. Davie. Computer Networks, A Systems Approach. Morgan Kaufman, Fourth edition, 2006. • A. S. Tanenbaum. Computer networks. Prentice-Hall, Fifth	Yes

	edition, 2010	
Recommended Texts	• James F. Kurose and KeithW. Ross. Computer Networking: A Top-Down Approach, Eighth edition, 2020	No
Websites	Jim Kurose Homepage (umass.edu)	

Grading Scheme

Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group **(50 - 100)**	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group **(0 – 49)**	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Microprocessor**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2104**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	1
Administering Department	Information Technology	**College**	College of Science
Module Leader	Aliabdulhussein ibrahim	**e-mail**	ali.abdulhussein19@uowa.edu.iq
Module Leader's Acad. Title	Asist. Lecturer	**Module Leader's Qualification**	MS.c
Module Tutor	Aliabdulhussein ibrahim	**e-mail**	ali.abdulhussein19@uowa.edu.iq
Peer Reviewer Name	Dr .Maky H.Abdulraheem	**e-mail**	maky.h@uowa.edu.iq
Scientific Committee Approval Date	2024-09-17	**Version Number**	V01

Relation with other Modules			
Prerequisite module	None	Semester	None
Co-requisites module	None	Semester	None

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د شيماء حسين نوفل
٢٠٢٤/٢٠٢٥

أ.م.د شيماء حسين نوفل
٢٠٢٤/٢٠٢٥

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	The purpose of the course is to teach and understand the main components and working principles of the 8086 processor. Understanding of basic computer architecture. Understanding memory organization and interaction with memory . Handling I/O units. The course analyzes the several components of a computing system: from the microprocessor internal architecture, up to system bus for peripheral devices management. The course also covers programming at assembly level.
Module Learning Outcomes	The course has following specific objectives: 1) Teaching the student about the microprocessor and its components and how to implement the instructions. 2) Learn assembly language 3) Knowing the methods and stages of converting an assembly language program into symbols. 4) Teaching the student, the principle of memory system and how it was divided the data into segments and how to link them. 5) To explain the principle of data flow.
Indicative Contents	Indicative content includes the following. ✓ Identifying the parts and components of the processor, the most important main units in the processor, identifying the memory, the most important signals that deal with it, and methods of accessing data inside the memory and the processor. ✓ A compiler design that uses specific algorithms in which data is entered in the manner of rules and laws are applied to it to know the results and to know the errors resulting in implementation and classify them according to their type and treat them

Learning and Teaching Strategies	
Strategies	The learning and teaching strategies for studying the microprocessor subject in an IT department involve a balanced approach of theoretical understanding and practical application. Lectures, interactive discussions, provide the necessary theoretical foundation. Practical exercises, group work, enable hands-on experience with microprocessor 8086. Giving lectures, carrying out assignments and practical issues inside the laboratories, conducting theoretical exams, discussions and scientific dialogues, and asking questions. These strategies ensure a comprehensive understanding of microprocessor and their relevance in the IT field.

Student Workload (SWL)			
Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	**6**
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	7	10% (8)	2,4,6,8,10	1,2,3,4
	Homework assignment	3	10% (7)	2,5,8,9,12	All
	Onsite Assignments	5	10% (5)	3,5,8,10,11	All
	Projects	1	10%(5)	12	All
	Lab	5	10%(10)	3,5,7,9,12	All
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Fundamental to microprocessor
Week 2	Microprocessor 8086 internal Architecture . Bus Interface Unit, Execution unit, register organization. Address bus, data bus, control bus
Week 3	
Week 4	Memory unit and R/W timing diagram memory segmentation Instruction cycle
Week 5	
Week 6	Memory addressing mode Instruction format
Week 7	
Week 8	Input / output devices and R/W timing diagram
Week 9	
Week 10	Assembly Language Programming Data transfer instruction set
Week 11	
Week 12	Variable , Array and constant
Week 13	Arithmetic and Logical instruction set
Week 14	
Week 15	Rotate and shift instruction set
Week 16	Preparatory week before the final Exam

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Lab 1: Setting up the emu8086 simulation
Week 2	Lab 2: the concept of Assembly Language
Week 3	Lab 3: Practical basic on assembly language
Week 4	Lab 4: learn to build a code using emu8086 simulation
Week 5	Lab 5,6 :Learn to create code for data transfer instruction set
Week 6	
Week 7	Lab 7,8: Learn to convert from Assembly language to machine language
Week 8	
Week 9	Lab 9,10: Learn to create code for arithmetic and logical instruction set
Week 10	
Week 11	Lab 11,12: Learn to deal with variable and array in emu8086 simulation
Week 12	
Week 13	Lab 13,14 : Learn to create code for rotate and shift instruction set
Week 14	
Week 15	Lab 15: implemented a code for preparing to the final exam

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	• The 80x86 Family, Design, Programming and Interfacing, 3rd edition,Prentice Hall, 2002.	
Recommended Texts	• The Intel Microprocessors, Architecture, Programming and Interfacing, Barry B. Brey, Prentice Hall, 1994.	

Grading Scheme

Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Object-oriented programming I**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2112**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	1
Administering Department	Information Technology	**College**	College of Science
Module Leader	Mohsin Hassan Hussein Abbas	**e-mail**	mohsin.ha@uowa.edu.iq
Module Leader's Acad. Title	Asst. Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Mohsen Hassan Hussein Abbas	**e-mail**	mohsin.ha@uowa.edu.iq
Peer Reviewer Name	Asst. Prof Haider Mohammed Ali	**e-mail**	hayder.alghanami@uowa.edu.iq
Scientific Committee Approval Date	2024-09-17	**Version Number**	V1.0

Relation with other Modules			
Pre-requisite module	Programming Fundamentals 2	**Semester**	2
Co-requisites module	Programming Fundamentals 2	**Semester**	2

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د. ضياء حسين نوفل
٢٠٢٥/٢٠٢٤

أ.م.د. ضياء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	1. Provide a sound knowledge of the underlying principles and experience in the practical application of this course is essential for any information technology specialist. 2. extend students with procedural programming knowledge and skills in the object-oriented paradigm and builds experience with interpreted languages to introduce compiled languages. 3. In addition to further shaping a solid development methodology, the course prepares students for continued investigation into advanced programming topics. 4. develop a wide range of software solutions for real-world scenarios.
Module Learning Outcomes	On completion of this course students will be able to: 1. identify and demonstrate an understanding of the hardware of a computer; 2. comprehend what programming is and what a programming language does; 3. know about the evolution of C++; 4. identify and design suitable classes and class hierarchies and code class implementations in C++; 5. design and develop C++ programs using classes and class libraries; 6. apply the principles of information hiding using C++ facilities for private and protected class attributes; 7. employ C++ facilities for dynamic storage; 8. employ C++ facilities such as operator overloading, pointers, and references; 9. develop programs using the C++ Standard for real-world.
Indicative Contents	**Topics**

	l scription	Weighting (75%)
1.	Overview of Object Oriented Programming, C++ or Python Basics	5.00
2.	Control flow	5.00
3.	Function Basics	5.00
4.	Parameters and Overloading	10.00
5.	Arrays and Structures	10.00
6.	Objects and Classes	10.00
7.	Constructors and Destructors	5.00
8.	Operator Overloading	5.00
9.	Friends and References	10.00
10.	Strings and Pointer	5.00
11.	Separate Compilation and Namespace	5.00

Learning and Teaching Strategies	
Strategies	**Overview Strategies** Object-oriented software development has become a standard methodology throughout the software engineering discipline. Therefore, a solid grasp of object-oriented programming is essential for any information technology specialist. While there are a variety of object-oriented programming languages available, C++ or Python are the most widely used in this course. This course extends the student's basic procedural design and programming knowledge and skills into the object-oriented paradigm and builds on previous experience with interpreted languages to introduce compiled languages. In addition to further shaping a solid development methodology, the course prepares students for continued investigation into advanced programming topics. The students will be expected to learn and apply the basic concepts of object oriented design and programming through giving lectures, practical exercises within the laboratories, assignments about some specific topics, and small projects. Key software engineering principles such as decomposition and component re-use will also be emphasized.

Student Workload (SWL)			
Structured SWL (h/sem)	75	**Structured SWL (h/w)**	5
Unstructured SWL (h/sem)	72	**Unstructured SWL (h/w)**	5
Student workload expectations (SWL &USWL) To do well in this subject, students are expected to commit approximately 10 hours per week including class contact hours, independent study, and all assessment tasks. If you are undertaking additional activities, the weekly workload hours may vary.			
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/ Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	5	10% (8)	3, ,6,9,11, 13	1,2,3,4
	OnSite Assignments	5	10% (5)	3,5,8,10,11	All
	HomeWork	5	10% (7)	2,5,8,10,12	All
	Project	1	10% (10)	12	All
	Labs	5	10% (15)	3,5,7,9,11	All
Summative assessment	Midterm Exam	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered	Weighting (30+5=35%)
Week 1	The fundamental concepts of programming, including procedural and object-oriented programming will be introduced. Also, consider the basic principles behind object-oriented programming techniques, including objects, classes, inheritance, and polymorphism. Then you will get started in programming environment by applying what you have learned.	2
Week 2	Introduction about the basic logic components used in programs that called control structures. It includes sequence structure, a selection structure, and loop structure, with examples.	2
Week 3	Learn about function features, including passing arguments, returning values, prototypes, and recursion, with examples.	2
Week 4	Present specific features of functions, such as function overloading and reference parameters, with examples.	2
Week 5	Introduce arrays concept with a specific element in an array, index, memory locations, the lowest address, highest address, arrays dimensions, arrays and pointers, with examples	2
Week 6	Overview about structures, structure declaration forms, and structure members, with examples.	2
Week 7	**Mid Term Exam Revision**	2
Week 8	Introduction about objects and classes, class declaration, Object declaration, with examples.	2
Week 9	Understanding constructors and destructors, constructors and destructors declaration with examples.	2
Week 10	Learn about overloading operators, operator declaration, unary operators, binary operators, and operator arguments.	2
Week 11	Learn what a friend is, Declare a friend function, and Examine the benefits of Use a friend function to access data from two classes, with examples.	2
Week 12	Understanding the three ways that a reference can be used: as a function parameter, as a function return value, or as a stand-alone reference, with examples.	2
Week 13	Learn about the string class , Learn about pointers, string and pointers declaration, with examples.	2

Week 14	Describes namespaces and several other advanced features, including conversion functions, explicit constructors, const and volatile member functions, the asm keyword, and linkage specifications, with examples.	2
Week 15	Students course workload evaluation.	2
Week 16	**Prepare to the final Exam**	3

Delivery Plan (Weekly Lab. Syllabus)		
	Material Covered	**Weighting (45%)**
Week 1 - Lab 1	- Prepare OOP environment, overview about unified modeling language (UML) diagram. - Access to a standard C++ or Python compiler - Linux g++ compiler and its equivalent MinGW running under windows.	3
Week 2 - Lab 2	- learn how to create a main () function, work with variables and constants, and create comments. - learn how to produce output and process input with Python or C++, and how to create first objects.	3
Week 3 - Lab 3	- Basic Functions and Pointers, - Implement recursion function, - Understand the manipulation on pointers.	3
Week 4 – Lab 4	- Understand function call by value method of parameter passing - Understand Pass parameters by reference method	3
Week 5 – Lab 5	- Study the use of structures - Understand array processing in C++ or Python - Understand heterogeneous data types	3
Week 6 – Lab 6	- Introduction to Classes and Objects	3
Week 7 – Lab 7	- Labs exam1 with evaluation	3
Week 8 – Lab 8	- Access Specifiers, Constructors and Destructors	3
Week 9 – Lab 9	- Constructor Overloading and Copy Constructors	3
Week 10 – Lab 10	- Introduction to Operator Overloading	3
Week 11 – Lab 11	- Friend Functions and Friend Classes	3
Week 12 – Lab 12	- Study string class and pointer concepts - Understand reference to an object concept	3

Week 13 – Lab 13	- Labs exam2 with evaluation	3
Week 14 – Lab 14	- Study the use of storage specifiers - Familiarise with global and static variables - Understanding separate Compilation and Namespace	3
Week 15 – Lab 15	- OOP project Implementation with discussion for each student	3

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	1. Malik, D.S 2018, *C++ Programming: Program Design Including Data Structures*, 8th edn, Cengage. (ISBN 978-1-337-11756-2.) 2. OOP – Learn Object Oriented Thinking and Programming, ISBN-10: 8090466184, Tomas Bruckner, 2013. 3. The student must have access to a standard C++ compiler. The only supported compilers are the Linux g++ compiler and its equivalent MinGW running under Windows.	No
Recommended Texts	4. Object-Oriented Programming Using C++ Fourth Edition by Joyce Farrell	No
Websites		

Grading Scheme

Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Principles of Database Systems**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2103**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	1
Administering Department	Information Technology	**College**	College of Science
Module Leader	Hussein Zaki Jassim	**e-mail**	hussein.almngoshi@uowa.edu.iq
Module Leader's Acad. Title	Asst. Lecturer	**Module Leader's Qualification**	Ph.D.
Module Tutor	Hussein Zaki Jassim	**e-mail**	hussein.almngoshi@uowa.edu.iq
Peer Reviewer Name	Asst. Prof Haider Mohammed	**e-mail**	hayder.alghanami@uowa.edu.iq
Scientific Committee Approval Date	2024-09-17	**Version Number**	V1.0

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

جامعة وارث الانبياء
عمادة كلية العلوم
وارث الانبياء جامعة
COLLEGE OF SCIENCES

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د. ضياء حسين نوئل
٢٠٢٤/٢٠٢٥

أ.م.د. ضياء حسين نوئل
٢٠٢٤/٢٠٢٥

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	1. Provide a solid understanding of database concepts, principles, and best practices. 2. Familiarize students with the design, implementation, and management of databases. 3. Cover topics such as data modeling, normalization, and query optimization. 4. Develop practical skills in using database management systems and query languages. 5. Cultivate critical thinking and problem-solving abilities in the context of database design and administration. 6. Prepare students to apply their knowledge in real-world scenarios. 7. Equip students to contribute to effective database solutions in the IT industry.
Module Learning Outcomes	1. Understand the fundamental concepts and principles of databases, including data models, schemas, and normalization. 2. Demonstrate proficiency in designing, implementing, and managing databases using a database management system (DBMS). 3. Apply data modeling techniques to develop logical and physical database designs that meet specified requirements. 4. Construct and execute complex SQL queries to retrieve, update, and manipulate data stored in a database. 5. Evaluate and optimize query performance through the use of indexing, query tuning, and other optimization techniques. 6. Implement and enforce data integrity constraints, including entity relationships, referential integrity, and data validation rules. 7. Employ appropriate security measures to protect data and ensure database confidentiality, integrity, and availability. 8. Utilize backup and recovery procedures to safeguard data and restore databases in the event of failures or disasters.
Indicative Contents	Indicative content includes the following. 1. Introduction: Briefly explain the purpose and goals of the database. Provide an overview of its intended users and stakeholders. Outline the benefits and value the database brings to the organization. 2. Database Design: Describe the overall structure and organization of the database.

	Identify the key entities, attributes, and relationships within the database. Explain the normalization process employed to ensure data integrity and eliminate redundancy. Discuss any design considerations specific to the database, such as performance optimization or scalability. 3. Data Model: Present the conceptual, logical, and physical data models used in the database. Explain the entity-relationship (ER) diagram, tables, and schema design. Discuss the various data types, constraints, and indexes used in the database. Highlight any additional modeling techniques or methodologies applied. 4. Functionality and Features: Enumerate the main functions and features provided by the database. Outline the CRUD operations (Create, Read, Update, Delete) supported. Describe any specialized or advanced features, such as data validation, triggers, or stored procedures. Mention any security measures implemented, such as user authentication and access control. 5. Data Sources and Integration: Identify the sources of data that feed into the database. Explain any data integration processes, including extraction, transformation, and loading (ETL). Discuss any data quality or cleansing procedures employed to ensure data accuracy. 6. Performance and Scalability: Discuss the database's performance characteristics, including response times and throughput. Describe any performance tuning techniques used, such as indexing or query optimization. Explain how the database handles scalability and growth, including considerations for increasing data volume or user load. 7. Maintenance and Administration: Outline the procedures for database backup, recovery, and disaster management. Explain the ongoing maintenance tasks, such as data archiving or purging. Describe the roles and responsibilities of database administrators. Mention any monitoring and alerting mechanisms in place.

Learning and Teaching Strategies	
Strategies	The learning and teaching strategies for sudying the database subject in an IT department involve a balanced approach of theoretical understanding and practical application. Lectures, interactive discussions, and case studies provide the necessary theoretical foundation. Practical exercises, group work, and projects enable hands-on experience with database management systems. Workshops, demos, and industry examples offer real-world insights. Online resources, assessments, and feedback aid in reinforcing learning. Virtual labs and continuous learning emphasize practical skills

	development and staying updated with industry trends. These strategies ensure a comprehensive understanding of databases and their relevance in the IT field.

Student Workload (SWL)

Structured SWL (h/sem	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/se	87	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	5	10% (8)	2,4,6,8,10	1,2,3,4,5,6,7
	OnSite Assignments	5	10% (5)	2,4,7,9,12	3,5,8
	Projects.	1	10% (7)	12	All
	Lab	5	10% (15)	3,5,7,9,11	All
	Home Work	5	10%(5)	2,5,8,9,12	All
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to databases: concepts, importance, and applications Relational database management systems (RDBMS)

Week 2	Overview: Introduction to Structured Query Language (SQL)
Week 3	Database design principles and data models
Week 4	Entity-Relationship (ER) modeling and ER diagrams
Week 5	Database constraints: primary key, foreign key
Week 6	Database constraints unique, and check constraints
Week 7	Database administration and security: user management, permissions, and access control
Week 8	Backup and recovery strategies for databases
Week 9	Indexing and query optimization techniques
Week 10	Transaction management and concurrency control in databases
Week 11	Relational model and relational calculus
Week 12	Relational model and relational algebra
Week 13	Transaction management and concurrency control in databases
Week 14	Transaction management and concurrency control in databases
Week 15	Database performance monitoring.
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Sy	
	Material Covered
Week 1	Lab 1: Setting up the database environment
Week 2	Lab 2: practicing basic SQL queries
Week 3	Lab 3: Designing an ER diagram for a given scenario
Week 4	Lab 4: translating ERD into a relational schema
Week 5	Lab 5: Normalizing a sample dataset and implementing the normalized tables in the database
Week 6	Lab 6: Learn all types of data used in database systems
Week 7	Lab 7: Learn to create a database with all specifications
Week 8	Lab 8: Learn to create tables with the ability to modify fields
Week 9	Lab 9: Learn addition operations for constraints in tables with constraints
Week 10	Lab 10: Learn operations for updates to constraints in tables with constraints
Week 11	Lab 11: Learn delete operations for constraints in tables with constraints
Week 12	Lab 12: Learn to build procedures for adding and modifying data

Week 13	Lab 13: Learn to build procedures with input variables
Week 14	Lab 14: Learn to construct procedures with output variables
Week 15	Lab 15: Implementation of an integrated database management project for each student

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	Elmasri, Ramez, and Shamkant Navathe. Fundamentals of database systems. AddisonWesley Publishing Company, 2018.	Yes
Recommended Texts	Database design, application and development.	No
Websites	http://www.sqlcourse.com/	

Grading Scheme				
Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Probability & Statistics**	**Module Delivery**	
Module Type	**Basic**	☒ **Lecture**	
Module Code	**IT2105**		
ECTS Credits	**4**		
SWL (hr/sem)	**100**		
Module Level	UG2	**Semester of Delivery**	1
Administering Department	Information Technology	**College**	College of Science
Module Leader	Ahmed Yahya Awad	**e-mail**	ahmed.ya@uowa.edu.iq
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	Ph.D.
Module Tutor	Ahmed Yahya Awad	**e-mail**	ahmed.ya@uowa.edu.iq
Peer Reviewer Name	Dr. Maky H.Abdulraheem	**e-mail**	maky.h@uowa.edu.iq
Scientific Committee Approval Date	2024-09-17	**Version Number**	V1.0

Relation with other Modules			
Prerequisite module	None	Semester	None
Co-requisites module	None	Semester	None

أ.م.د شيماء حسين نوفل
٢٠٢٤/٢٠٢٥

جامعة وارث الانبياء
عمادة كلية العلوم
وارث الانبياء جامعة

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د شيماء حسين نوفل
٢٠٢٤/٢٠٢٥

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	1- This module will provide students with a basic knowledge of mathematical probability theory and the techniques of statistical inference that are used for analyzing data. 2- Also, this module will provide students a foundation for further modules in statistics and applied probability. 3- Understanding the most important principles of statistics and statistical methods for representing data, as well as knowing the types of coefficients statistics, their importance and methods of calculation. 4- Understanding the most important principles of probability and the most important operations that take place on the aggregates and knowing what most important properties of probability.
Module Learning Outcomes	On successful completion of this module, a student will be able to: 1- Model simple experiments using probability theory. 2- Perform standard probability calculations. 3- Work with independent and correlated random variables. 4- Correctly apply simple formal statistical techniques and interpret the results. 5- Assess, analyses and interpret basic statistical problems. 6- Discern when statistics are being misused. 7- Present results of basic statistical analyses (both descriptive and inferential). 8- Apply simple probabilistic and statistical concepts. 9- Construct and apply mathematical descriptions of probability distributions.
Indicative Contents	1. Introduction to Probability Theory ○ Basic concepts of probability: sample spaces, events, and probability axioms. ○ Combinatorial principles and counting techniques. ○ Conditional probability and independence. ○ Discrete and continuous probability distributions. ○ Expected value, variance, and moment-generating functions. 2. Statistical Data Representation ○ Data types: qualitative and quantitative. ○ Graphical representation of data: histograms, bar charts, and pie charts. ○ Measures of central tendency: mean, median, and mode. ○ Measures of dispersion: range, variance, and standard deviation. ○ Exploratory data analysis techniques. 3. Statistical Inference ○ Sampling techniques and sampling distributions. ○ Point estimation: methods for estimating population parameters. ○ Interval estimation: construction of confidence intervals. ○ Hypothesis testing: formulation of null and alternative hypotheses, test statistics, and p-values. ○ Type I and Type II errors, significance level, and power of tests.

	4. Probability Distributions ○ Binomial, Poisson, and normal distributions: properties and applications. ○ Central Limit Theorem and its significance. ○ Transformations of random variables. ○ Joint probability distributions and independence. ○ Multivariate distributions: covariance, correlation, and regression. 5. Statistical Methods and Techniques ○ Regression analysis: simple linear regression and multiple regression. ○ Analysis of variance (ANOVA): one-way and two-way ANOVA. ○ Nonparametric methods: rank tests and chi-square tests. ○ Experimental design and sampling strategies. ○ Data collection, validation, and interpretation. 6. Foundations for Further Study in Statistics and Applied Probability ○ Bridging concepts and techniques for more advanced statistical modules. ○ Connecting probability theory and statistical inference to real-world applications. ○ Understanding the importance of statistical methods in decision-making and research.

Learning and Teaching Strategies	
Strategies	1- Giving weekly lecture/tutorial sessions. 2- Printed notes will be given for each part of the course. 3- Concepts and underlying theories will be explored in the lecture period. 4- Students will learn through a formative process of tackling the exercises at the end of each section, with feedback and extension in tutorials. 5- Scientific discussions and dialogues and asking questions.

Student Workload (SWL)			
Structured SWL (h/sem)	45	**Structured SWL (h/w)**	3
Unstructured SWL (h/sem)	52	**Unstructured SWL (h/w)**	3.5
Total SWL (h/sem)	97 + 3 final = 100		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	5	10% (10)	3,6,9	1,2,3,4
	Onsite Assignments	5	10% (10)	2,4,6,10,12	All
	H.W	5	10% (10)	2,4,6,8,10	All
	Report	1	10% (10)	12	All
Summative assessment	Midterm Exam	2hr	10% (10)	5,11	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Probability (Sample Space, Events, Probability of an Event)
Week 2	Probability (Additive Rules, Independence, Product Rule)
Week 3	Conditional Probability
Week 4	Total Probability Rule.
Week 5	Bayes’ Rule.
Week 6	Discrete and Continuous Random Variable.
Week 7	Probability Density Functions.
Week 8	Joint Probability Distributions.
Week 9	Probability Mass Functions.
Week 10	Cumulative Distribution Functions.
Week 11	Statistics Basics
Week 12	Frequency Distributions
Week 13	Measures of Central Tendency
Week 14	Discrete Uniform Distribution.
Week 15	Measures of Dispersion

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	1. An introduction to probability and statistics. (R1) 2. Introduction to Statistics. (R2)	

Grading Scheme

Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Professional Ethics**	**Module Delivery**	
Module Type	**Supportive**	☒ **Lecture**	
Module Code	**IT2106**		
ECTS Credits	**2**		
SWL (hr/sem)	**50**		
Module Level	UG2	**Semester of Delivery**	1
Administering Department	Information Technology	**College**	College of Science
Module Leader	Haider Mohammed Ali Ali Al-Ghanim	**e-mail**	hayder.alghananmi@uowa.edu.iq
Module Leader's Acad. Title	Asst. Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Haider Mohammed Ali Ali Al-Ghanim	**e-mail**	hayder.alghananmi@uowa.edu.iq
Peer Reviewer Name	Asst. Lecturer Karrar Sadiq	**e-mail**	karar.sadeq@uowa.edu.iq
Scientific Committee Approval Date	2024-09-17	**Version Number**	V1.0

Relation with other Modules

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

أ.م.د. شيماء حسين نوبل
٢٠٢٤/٢٠٢٥

جامعة وارث الانبياء
عمادة كلية العلوم
COLLEGE OF SCIENCES

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د. شيماء حسين نوبل
٢٠٢٤/٢٠٢٥

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	This module aims to provide students with a comprehensive understanding of computer ethics and the social and ethical considerations associated with the world of information technology. It focuses on developing the necessary skills to analyze problems, research current ethical issues in information systems and the Internet, and apply ethical principles and best practices in the field of information technology. By the end of the module, students should be able to identify ethical challenges and make informed decisions to address them, utilizing modern, ethical, and socially responsible approaches to promote positive outcomes and mitigate potential risks.
Module Learning Outcomes	1. The ability to identify computer ethics and social and ethical issues to be followed in the world of information technology. 2. The ability to analyze problems and determine the means required for a solution. 3. The ability to research and study the latest findings of the world in the field of ethics of information systems and the Internet, which is of great importance in our world today and in our private and public societies. 4. Understanding the procedures supporting computer ethics, trying to apply them, and finding the best solutions for them. 5. The ability to apply the best modern, ethical, and social ways in the field of information technology, benefit from the positive aspects, and avoid the negative matters and problems that constitute the most dangerous currents for all humanity.
Indicative Contents	Indicative content includes the following. 1. Demonstrate knowledge of current models of information and computer ethics. 2. Apply ethical theories to interpret personal and group when using a variety of information technology tools. 3. Evaluate the nature of ethical choices made by self and others when serving various roles that expose social and multicultural differences. 4. Construct written arguments in a variety of formats on the evolving nature of ethical norms relating to new technologies. 5. Prepare and deliver an oral presentation for a user audience. 6. Prepare and deliver an oral presentation for a management audience. 7. Write a technical memo to management. 8. Create user documentation for an IT system. 9. Create a set of technical requirements for an IT system. 10. Compare and contrast technical writing and expository writing.

Learning and Teaching Strategies	
Strategies	The learning and teaching strategies for studying the Professional Ethics subject in the IT department involve: ✓ Lectures. ✓ Interactive discussion. ✓ Assessments which include individual assignments, quizzes, and examinations. ✓ Provide the necessary theoretical foundation. ✓ Online resources, and feedback aid in reinforcing learning for humans, like expressing their own values, exploring, with empathy, the values of others, critically analyse values and actions based on them, discussing disagreements that arise from differences in values, and negotiate solutions, make ethical decisions, and act on them. These strategies ensure a comprehensive understanding of Professional Ethics and its relevance in the IT field.

Student Workload (SWL)			
Structured SWL (h/sem)	30	**Structured SWL (h/w)**	2
Unstructured SWL (h/sem)	17	**Unstructured SWL (h/w)**	1.2
Total SWL (h/sem)	47 + 3 final = 50		

Module Evaluation					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	5	10% (15)	4,6,8,10,12	All
	Assignments	2	10%(5)	4,8	All
	H.W	3	10% (10)	3,5,10	All
	Report	1	10% (10)	12	All
Summative assessment	**Midterm Exam**	2hr	10% (10)	5,11	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)	
	Material Covered
Week 1	General Introduction to the field of computer and information ethics. Explain the History as the information age and The philosophy of information
Week 2	Computer Ethics and Information Ethics.
Week 3	Explain the Classical Computer ethics theory and Computer ethics with new challenges.
Week 4	Ethical issues in Information Technology.
Week 5	Explain the Ethical issues in Information Technology, why it is important in information technology?, Rights and computer ethics.
Week 6	Ethical issues in Information Technology. Personal values and computer ethics, Different Views on Ethical Behaviour.
Week 7	Ethics and the Professions. Explain the Origins of Professions, Professional requirements, a professional behaves. ethically, Professionalism in Information Technology.
Week 8	IT Professionals. Explain what is IT Professionals, Why IT professionalism is needed and why is it important?
Week 9	Explain the Role of Ethics and Professionalism in IT, the Professional and non-Professional.
Week 10	Ethics for IT Workers and IT Users. Discuss IT Professional and IT Users.
Week 11	Intellectual property crime Intellectual Property Rights and Computer Technology, Infringement.
Week 12	The Ownership. Explain the politics of ownership.
Week 13	Software Development Explain the Software Product Liability, Key Issues in software development
Week 14	Software Development Challenges. Explain the Challenges in Software Development, Challenges of Software Product Developer.
Week 15	Computer Security. Explain computer and Internet Crime, Define Hacking and Hacktivism.
Week 16	**Preparatory week before the Final Exam**

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	Ethics in Information Technology; Reynolds, George, ASIN, 1337405876; Publisher, Cengage Learning; 6th edition (January 1, 2018); Language, English; Paperback, 480 pages.	
Recommended Texts	Ethical and Social Issues in the Information Age by Joseph Migga Kizza	
Websites		

Grading Scheme

Group	Grade	Mark	Marks (%)	Definition
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Arabic Language 2**	**Module Delivery**	
Module Type	**Supportive**	**Lecture**	
Module Code	**UOWA203**		
ECTS Credits	**2**		
SWL (hr/sem)	**50**		
Module Level	UG2	**Semester of Delivery**	2
Administering Department	Information Technology	**College**	Sciences
Module Leader	Ayad Karim	**e-mail**	ayadalsalahi@uowa.edu.iq
Module Leader's Acad. Title	Professor Doctor	**Module Leader's Qualification**	Ph.D
Module Tutor	Ayad Karim	**e-mail**	ayadalsalahi@uowa.edu.iq
Peer Reviewer Name	Asst.Lect Nabeel Sadeq Al-Shreefy	**e-mail**	nabeel.alshreefy@uowa.edu.iq
Scientific Committee Approval Date	2025-01-28	**Version Number**	V1

Relation with other Modules

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

أ.م.د ضياء حسين نوفل
٢٠٢٥/٢٠٢٤

جامعة وارث الانبياء
عمادة كلية العلوم
جامعة وارث الانبياء
COLLEGE OF SCIENCES

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د ضياء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	**Course Objectives** The objectives of studying the Arabic language course focus on introducing students to the most important basic rules of correct spelling, avoiding errors in expression, and mastering them in a way that suits the student's cultural level. Here are some of the main objectives of the course: 1. To understand the basic rules of orthography that students need for their studies and future work. 2. To be able to apply these rules easily and smoothly without needing to memorize them by heart. 3. To enable students to recognize simple spelling and language mistakes to avoid making them. 4. To develop the ability to express oneself correctly without relying on others. 5. To make students aware of the importance of the Arabic language in their life and work, expand their linguistic knowledge, identify common errors and difficulties they face, and learn how to address them.
Module Learning Outcomes	**Learning Outcomes** The learning outcomes of the Arabic language course aim to develop a set of skills and knowledge among students. Below are some of the main learning outcomes expected upon completing this course: 1. To explain the basic rules of writing in the Arabic language. 2. To apply linguistic knowledge skillfully and effortlessly. 3. To analyze simple linguistic structures and texts used in daily life. 4. To demonstrate self-confidence and the ability to express oneself easily. 5. To correct and address simple language errors independently.
Indicative Contents	1. **Promoting the Arabic language** among members of society in general to open new horizons for developing and enhancing linguistic skills. 2. **Addressing the various challenges** facing society in the fields of education and upbringing, especially in teaching the Arabic language, and striving to find effective and useful solutions to strengthen language abilities. 3. **Utilizing modern communication tools**, such as the Internet and others, in the learning process.

Learning and Teaching Strategies	
Strategies	Many simple strategies can be used to effectively develop the learning process and make it enjoyable and beneficial, including: 1. **Numbered Heads Strategy** 2. **Ice Cream Sticks Strategy** 3. **Think-Pair-Share Strategy** 4. **Cube Strategy** 5. **Error Correction Strategy** 6. **Hot Seat Strategy**

Student Workload (SWL)			
Structured SWL (h/sem)	26	**Structured SWL (h/w)**	1.75
Unstructured SWL (h/sem)	21	**Unstructured SWL (h/w)**	1.4
Total SWL (h/sem)	47 + 3 Final Exam = 50		

Module Evaluation					
		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	3	10% (11)	2,3,4	3,4
	Home Work	2	10% (10)	All Weeks	2,3,4
	Onsite Assignments	1	10% (8)	All Weeks	1,2,3,4,5
	Reports	1	10% (11)	All Weeks	1,2,3,4,5
Summative assessment	**Midterm Exam**	2hr	10% (10)	9	
	Final Exam	3hr	50% (50)	17	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introductory Overview of Arabic Language Studies
Week 2	Parts of Speech – Nouns and Particles
Week 3	Parts of Speech – Verbs
Week 4	Grammatical Case Endings (I‘rab)
Week 5	Singular and Dual Forms
Week 6	Plurals and Their Types
Week 7	Verbal Sentences
Week 8	Nominal Sentences
Week 9	Midterm Exam
Week 10	“Kana” and Its Sisters
Week 11	“Inna” and Its Sisters
Week 12	Nominative, Accusative, and Genitive Cases
Week 13	Nominative, Accusative, and Genitive Cases – Part 2
Week 14	Literary Periods
Week 15	Literary Periods – Part 2
Week 16	Preparation for the Final Exam

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	شرح ابن عقيل، تحقيق محمد محي الدين عبد الحميد ، دار إحياء التراث ، بيروت لبنان جامع الدروس العربية، مصطفى الغلاييني، دار الثقلين، بيروت لبنان. الأدب العربي، ناظم رشيد	
Recommended Texts		
Websites	مكتبة لسان العرب الإلكترونية شبكة الألوكة موقع فصيح مكتبة نرجس الإلكترونية المكتبة الوقفية الإلكترونية مكتبة نور الإلكترونية	

Grading Scheme

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

Module Information			
Module Title	Crimes of the defunct Baath Party	**Module Delivery**	
Module Type	S SUPPORTIVE	Lecture	
Module Code	UOWA104		
ECTS Credits	2		
SWL (hr/sem)	50		
Module Level	UG2	**Semester of Delivery**	2
Administering Department	IT	College	Science
Module Leader	Asst. Lecturer Abbas Taher	**e-mail**	Abbas.taher@uowa.edu.iq
Module Leader's Acad. Title	Asst. Lecturer	**Module Leader's Qualification**	MS.c
Module Tutor	Asst. Lecturer Abbas Taher	**e-mail**	Abbas.taher@uowa.edu.iq
Peer Reviewer Name	Asst. Lecturer Karrar Sadiq	**e-mail**	karar.sadeq@uowa.edu.iq
Review Committee Approval	2025-02-01	**Version Number**	V1

Relation With Other Modules			
Prerequisite module	None	**Semester**	None
Co-requisites module	None	**Semester**	None

أ.م.د ضياء حسين نوفل
٢٠٢٤/٢٠٢٥

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د ضياء حسين نوفل
٢٠٢٤/٢٠٢٥

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	• Understand the concept of crime and its elements. • Learn about international and local laws that criminalize the actions committed by the Ba’ath regime in Iraq. • Identify the most prominent cases and crimes addressed by the Iraqi High Criminal Court and understand the scale of these crimes. • Recognize the psychological and social effects on citizens, including the resulting scientific and cultural impoverishment. • Examine the environmental impacts, including public health deterioration due to war remnants and marsh drainage. • Identify the mass graves left by the Ba’ath regime, specifying the location and time of each grave.
Module Learning Outcomes	1- Understand the definition of crime, its definitions in other sciences, and its classifications. 2- Identify the most prominent cases addressed by the Iraqi High Criminal Court, the verdicts issued against the convicted, and the laws under which death sentences were handed down to Ba’ath regime affiliates. 3- Recognize the negative effects and harsh methods used in the militarization of society. 4- Recall the most prominent religious figures who were persecuted and arrested due to their stance and opposition to the Ba’ath regime. 5- Identify major political violations and raise awareness among community members about the Ba’ath regime’s violations and their impact on Iraqi society. 6- Highlight secret prisons and private detention centers where opponents of the Ba’ath regime were imprisoned. 7- Explain the most significant environmental violations committed by the Ba’ath regime in Iraq. 8- Describe and explain areas of radiation pollution and how the regime used the scorched-earth policy against villages and cities. 9- Explain the process of draining the marshes in southern Iraq and uprooting orchards and palm trees following the 1991 Shia uprising. 10- Identify the mass graves committed by the Ba’ath regime. 11- Explain the events of 1963, including the killings under Abdul Karim Qasim’s regime, and their connection to mass graves. 12- Classify mass graves during the Iran-Iraq war and the associated Anfal massacre events of 1987–1988. 13- Examine the documentation of the 1991 Shia uprising and the mass graves created by the Ba’ath regime after suppressing participants. 14- Present mass graves according to the dates of occurrence from 1963 until the fall of the regime in 2003, including the Barzani Kurds’ grave (1983), Anfal massacre graves (1987–1988), and the victims of the 1991 Shia uprising.

Indicative Contents	1- Concept of Crime and Its Types (3 hours) 2- Crimes of the Ba'ath Regime According to the Documentation of the Iraqi High Criminal Court Law 2005 (2 hours) 3- Decisions Issued by the Iraqi High Criminal Court (2 hours) 4- Psychological and Social Crimes and Their Effects (2 hours) 5- Militarization of Society (2 hours) 6- The Ba'ath Regime's Position on Religion and Violations of Iraqi Laws (2 hours) 7- Human Rights Violations and Crimes of Authority (2 hours) 8- Some Political and Military Violation Decisions of the Ba'ath Regime (2 hours) 9- Environmental Crimes of the Ba'ath Regime in Iraq (2 hours) 10- War and Radiation Pollution, Mine Explosions, and Destruction of Cities and Villages (3 hours) 11- Draining of the Marshes in Southern Iraq and Uprooting of Orchards and Crops (2 hours) 12- Crimes of Mass Graves (2 hours) 13- Events of Mass Graves Committed by the Ba'athist Regime in Iraq (2 hours) 14- Chronological Classification of Mass Graves in Iraq from 1963–2003 (2 hours)

Learning and Teaching Strategies

Strategies	Interactive Lecture Brainstorming Dialogue and Discussion Self-Learning

Student Workload (SWL)

Structured SWL (h/sem)	26	**Structured SWL (h/w)**	1.75
Unstructured SWL (h/sem)	21	**Unstructured SWL (h/w)**	1.4
Total SWL (h/sem)	47 + 3 final = 50		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	3	10% (11)	All Weeks	1-10
	Home Work	2	10% (10)	3, 10	1,2,3,4,5,6,7,8,9
	Assignments in College	1	10% (8)	all	1-12
	Report	1	10% (11)	All Weeks	1-14
Summative assessment	Midterm Exam	1	10	7	1-6
	Final Exam	1	50	16	all
Total assessment			100		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Concept of Crimes and Their Categories
Week 2	Crimes of the Ba'ath Regime According to the Documentation of the Iraqi Criminal Court Law of 2005
Week 3	Decisions of the Iraqi High Criminal Court
Week 4	Mechanisms of Psychological Crimes
Week 5	Militarization of Society
Week 6	The Ba'ath Regime's Stance on Religion
Week 7	Midterm Exam
Week 8	Violations of Iraqi Laws
Week 9	Environmental Crimes of the Ba'ath Regime in Iraq
Week 10	Draining of Marshes and Uprooting of Orchards and Crops
Week 11	Crimes of Mass Graves
Week 12	Events of Mass Graves Committed by the Ba'athist Regime in Iraq
Week 13-14	Chronological Classification of Mass Graves and Genocide in Iraq (1963–2003)
Week 15	Mass Graves Committed by the Former Ba'ath Regime During the Period 1979–2003
Week 16	Final Exam

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	1- جرائم نظام حزب البعث(منهاج وزاري) 2- د. احمد خالد عبد القادر، حرب الابادة الجماعية ضد الشعب العراقي والقانون الدولي المعاصر. 3- د. حمدي ابو النجار، مخاطر التلوث البيئي.	نعم
Recommended Texts	مجلة حقوق الانسان والحريات العامة. أنطونيو كاسيزي، القانون الجنائي الدولي.	نعم
Websites	المركز العراقي لتوثيق جرائم التطرف https://iraqicenter-fdec.org/archives/5146	

APPENDIX:

GRADING SCHEME

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

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

ملاحظة: هذا النموذج تم وضعه وتقديمه من قبل مديرية ضمان الجودة في وزارة التعليم العالي والبحث العلمي

MODULE DESCRIPTION FORM

Module Information		
Module Title	**Data Structure**	**Module Delivery**
Module Type	Core	**Lecture** **Lab** **Practical**
Module Code	**IT2204**	
ECTS Credits	6	
SWL (hr/sem)	150	

Module Level	UG2	**Semester of Delivery**	2
Administering Department	Information Technology	**College**	College of computer science and information technology
Module Leader	Karar Sadiq Mohsen Al-Ghadhri	**e-mail**	karar.sadeq@uowa.edu.iq
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	M.Sc.
Module Tutor	Karar Sadiq Mohsen Al-Ghadhri	**e-mail**	karar.sadeq@uowa.edu.iq
Peer Reviewer Name	Asst. Prof Haider Mohammed	**e-mail**	hayder.alghanami@uowa.edu.iq
Scientific Committee Approval Date	2025-01-28	**Version Number**	1

Relation with other Modules			
Prerequisite module	Programming Fundamentals 2	**Semester**	2
Co-requisites module	-	**Semester**	-

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Objectives	1. Understand the general principle of data structures and their usefulness 2. Knowing the types of data structures and the main principle of each structure 3. Knowledge of applications related to the subject and other fields of knowledge
Module Learning Outcomes	1. Understanding Fundamental Data Structures: The course aims to provide students with a solid understanding of fundamental data structures such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Students learn about the characteristics, operations, and implementation details of these data structures. 2. Implementing Data Structures: The course focuses on teaching students how to implement various data structures in a programming language of their choice. This hands-on experience helps students gain practical skills in implementing and manipulating data structures efficiently. 3. Problem Solving: Data Structures course typically includes problem-solving exercises and assignments. Students are presented with real-world problems and learn how to model them using appropriate data structures. They develop problem-solving skills by designing algorithms and implementing solutions using the learned data structures.
	4. Memory Management: Understanding data structures also involves understanding how memory is managed in a program. Students learn about concepts such as dynamic memory allocation, pointers, memory leaks, and memory optimization techniques to ensure efficient use of memory in data structures. 5. Algorithmic Thinking and Efficiency: The course cultivates algorithmic thinking skills in students. They learn how to break down complex problems into smaller, solvable tasks and design efficient algorithms using appropriate data structures. Emphasis is placed on optimizing algorithms for time and space efficiency.

Indicative Contents	In Data Structures, there are several key components that should be considered. While the specific contents may vary depending on the nature of the structure and the context in which it is being used, the basic contents in Data Structures are: Part A – Data structures principles 1. Introduction to Data Structures: Overview of data structures, their importance, and their role in problem-solving. Introduction to the basic terminology and concepts related to data structures (collections) [10 hrs] 2. Arrays: Study of arrays as a fundamental data structure. Understanding the concept of indexing, accessing, and manipulating array elements. Exploring multidimensional arrays and their applications. [15 hrs] Part B – Linear Structures 1. Stacks: Understanding the stack data structure and its LIFO (Last-In-First-Out) behavior. Implementing stack operations, such as push, pop, and peek. Applications of stacks, such as expression evaluation, function call management, and backtracking . [15 hrs] 2. Queues: Introduction to queues and their FIFO (First-In-First-Out) behavior. Implementing queue operations, such as enqueue and dequeue. Exploring different types of queues, including circular queues and priority queues.

Applications of queues in real-world scenarios [15 hrs]

Part C – Nonlinear Structures

1. Linked Lists: Introduction to linked lists as a dynamic data structure. Understanding the different types of linked lists, such as singly linked lists, doubly linked lists, and circular linked lists. Operations on linked lists, including insertion, deletion, traversal, and searching. [15 hrs]
2. Trees: Study of tree structures, including binary trees, binary search trees, AVL trees, and B-trees. Understanding tree traversal algorithms (pre-order, in-order, post-order) and operations such as insertion, deletion, and searching in trees. Applications of trees, such as hierarchical data representation and sorting [15 hrs]
3. searching algorithms (e.g., binary search) [15 hrs]
4. graph algorithms (e.g., breadth-first search, depth-first search) [15 hrs]
5. Hash Tables: Understanding the concept of hashing and hash functions. Exploring hash table data structure and collision resolution techniques (chaining, open addressing). Implementing hash table operations, such as insertion, deletion, and searching. Applications of hash tables, such as dictionary implementations and efficient lookup. [15 hrs]
6. Recursion: Understanding the concept of recursion and its applications in data structures and algorithms. [15 hrs]

Learning and Teaching Strategies	
Strategies	Throughout the module, it is important to employ effective learning and teaching strategies to ensure students grasp the concepts and develop the necessary skills. The main strategy that will be adopted in delivering this module: Giving lectures. Practical Exercises within the laboratories: Provide students with opportunities to practice their skills through practical exercises. Conducting theoretical exams. Request a report from students to teach them how to do research on a specific topic. Request some small projects from student groups of real-world examples and case studies to demonstrate the relevance and impact of different Data structures

Student Workload (SWL)			
Structured SWL (h/sem)	60	Structured SWL (h/w)	4
Unstructured SWL (h/sem)	87	Unstructured SWL (h/w)	5
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	5	10% (8)	All Weeks	5,10
	Onsite Assignments	5	10% (7)	All Weeks	6
	Lab	5	10% (15)	All Weeks	1,2,3,4,5,6,7,8,9,10
	Home Work	5	10% (7)	All Weeks	6
	Project	1	10% (5)	14	6,15
Summative assessment	Midterm Exam	2hr	10% (10)	8	
	Final Exam	3hr	50% (50)	17	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to Data structures
Week 2	Arrays, structures, pointers, memory allocation
Week 3	Stack and their basic operations.
Week 4	Queue and their basic operations.
Week 5	Recursion

Week 6	Single linked list
Week 7	double linked list
Week 8	circular linked list
Week 9	circular linked list 2
Week 10	Trees, binary search trees and basic operations
Week 11	Graphs and basic algorithms on graphs: depth first and breadth first search
Week 12	Sorting Algorithms
Week 13	Searching Algorithms
Week 14	Dijkstra’s algorithm.
Week 15	Priority queues
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Lab 1: Programming fundamentals overview
Week 2	Lab 2: Arrays, structures, pointers, memory allocation
Week 3	Lab 3: Stack and their basic operations.
Week 4	Lab 4: Queue and their basic operations.
Week 5	Lab 5: Recursion
Week 6	Lab 6: Single linked list
Week 7	Lab 7: double linked list
Week 8	**Lab 8: double linked list 2**

Week 9	Lab 9: circular linked list
Week 10	Lab 10: Trees, binary search trees and basic operations
Week 11	Lab 12: Graphs and basic algorithms on graphs: depth first and breadth first search
Week 12	Lab 14: Sorting Algorithms
Week 13	Lab 15: Searching ALgorithms
Week 14	Lab 13: Dijkstra's algorithm. Priority queues
Week 15	**Project Submission**

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	1. Clifford A. Shaffer (2009). A Practical Introduction to Data Structures and Algorithm Analysis (3rd), Prentice Hall 2. MICHAEL MCMILLAN. (2007). DATA STRUCTURES AND ALGORITHMS USING C#. Michael McMillan 2007 3. Granville Barnett, and Luca Del Tongo (2008). Data Structures and Algorithms: Annotated Reference with Examples (1st)	No
Recommended Texts		
Websites		

Grading Scheme

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

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

MODULE DESCRIPTION FORM

Module Information			
Module Title	**Database Systems Design and Development**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2203**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	2
Administering Department	Information Technology	**College**	College of Science
Module Leader	Hussein Zaki Jassim Al-Mankoushi	**e-mail**	hussein@uowa.edu.iq
Module Leader's Acad. Title	Asst. Lect	**Module Leader's Qualification**	M.Sc.
Module Tutor	Hussein Zaki Jassim Al-Mankoushi	**e-mail**	hussein@uowa.edu.iq
Peer Reviewer Name	Asst. Prof Haider Mohammed	**e-mail**	hayder.alghanami@uowa.edu.iq
Scientific Committee Approval Date	2025-01-20	**Version Number**	1.0

Relation with other Modules			
Prerequisite module	Principles of database systems	**Semester**	1
Co-requisites module	Principles of database systems	**Semester**	1

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د. شيماء حسين نوفل
٢٠٢٤/٢٠٢٥

أ.م.د. شيماء حسين نوفل
٢٠٢٤/٢٠٢٥

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	1. Provide a solid understanding of database concepts, principles, and best practices. 2. Familiarize students with the design, implementation, and management of databases. 3. Cover topics such as data modeling, normalization, and query optimization. 4. Develop practical skills in using database management systems and query languages. 5. Cultivate critical thinking and problem-solving abilities in the context of database design and administration. 6. Prepare students to apply their knowledge in real-world scenarios. 7. Equip students to contribute to effective database solutions in the IT industry.
Module Learning Outcomes	1. Understand the fundamental concepts and principles of databases, including data models, schemas, and normalization. 2. Demonstrate proficiency in designing, implementing, and managing databases using a database management system (DBMS). 3. Apply data modeling techniques to develop logical and physical database designs that meet specified requirements. 4. Construct and execute complex SQL queries to retrieve, update, and manipulate data stored in a database. 5. Evaluate and optimize query performance through the use of indexing, query tuning, and other optimization techniques. 6. Implement and enforce data integrity constraints, including entity relationships, referential integrity, and data validation rules. 7. Employ appropriate security measures to protect data and ensure database confidentiality, integrity, and availability. 8. Utilize backup and recovery procedures to safeguard data and restore databases in the event of failures or disasters.
Indicative Contents	Indicative content includes the following. 1. Advanced Database Design: Entity-Relationship Modeling: Extensions and enhancements to ER modeling, such as subtypes, supertypes, and specialization/generalization. Object-Oriented Data Modeling: Concepts of object-oriented databases and their modeling techniques, including inheritance, encapsulation, and polymorphism.

	UML Diagrams: Utilizing Unified Modeling Language (UML) to model databases, including class diagrams, object diagrams, and sequence diagrams. 2. Transaction Management and Concurrency Control: ACID Properties: Understanding the properties of atomicity, consistency, isolation, and durability in database transactions. Concurrency Control: Techniques for managing concurrent access to the database, including locking, timestamp-based protocols, and optimistic concurrency control. Recovery and Undo/Redo Logging: Mechanisms for ensuring database consistency in the face of failures, including log-based recovery and transaction rollback/commit. 3. Query Optimization and Execution: Query Processing: The stages involved in processing a database query, including parsing, optimization, and execution. Query Optimization: Techniques for selecting the most efficient query execution plan, such as cost-based optimization, join ordering, and index selection. Query Execution: Strategies for executing queries, including algorithms for sorting, joining, and aggregating data. 4. Data Storage and Indexing: File Structures: Storage structures for database files, such as heap files, sorted files, and hashed files. Indexing Techniques: Different indexing structures for efficient data retrieval, including B-trees, hash indexes, and bitmap indexes. Multi-Dimensional Data Structures: Introduction to data structures like R-trees and quad-trees for indexing spatial and multidimensional data. 5. Database Security and Authorization: Database Security: Concepts of access control, authentication, and authorization in database systems. Security Models: Different security models, such as discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC). Encryption and Auditing: Techniques for encrypting data and auditing database activities for security and compliance purposes.

Learning and Teaching Strategies	
Strategies	The learning and teaching strategies for studying the database subject in an IT department involve a balanced approach of theoretical understanding and practical application. Lectures, interactive discussions, and case studies provide the necessary theoretical foundation. Practical exercises, group work, and projects enable hands-on experience with database management systems. Workshops, demos, and industry examples offer real-world insights. Online resources, assessments, and feedback aid in reinforcing learning. Virtual labs and continuous learning emphasize practical skills development and staying updated with industry trends. These strategies ensure a comprehensive understanding of databases and their relevance in the IT field.

Student Workload (SWL)

Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	5	10% (8)	2,3,4,5,6,7	All Outcome
	Onsite Assignments	5	10% (5)	All Weeks	All Outcome
	Lab	5	10% (15)	All Weeks	All Outcome
	Projects	1	10% (7)	All Weeks	All Outcome
	HW	5	10% (5)	All Weeks	All Outcome
	Report	1	10% (10)	13	
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to databases: concepts, importance, and applications Relational database management systems (RDBMS)
Week 2	Overview: Introduction to Structured Query Language (SQL)
Week 3	Database design principles and data models

Week 4	Entity-Relationship (ER) modeling and ER diagrams
Week 5	Database constraints: primary key, foreign key
Week 6	Database constraints unique, and check constraints
Week 7	Database administration and security: user management, permissions, and access control
Week 8	Backup and recovery strategies for databases
Week 9	Indexing and query optimization techniques
Week 10	Transaction management and concurrency control in databases
Week 11	Relational model and relational calculus
Week 12	Relational model and relational algebra
Week 13	Transaction management and concurrency control in databases
Week 14	Transaction management and concurrency control in databases
Week 15	Database performance monitoring.
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Set up a database environment using a preferred database management system
Week 2	Create tables in the database based on the schema design
Week 3	Populate the tables with sample data to simulate real-world scenarios. Include a sufficient amount of data to perform meaningful queries.
Week 4	SELECT Queries: Write and execute basic SELECT queries to retrieve data from single tables.
Week 5	Use various clauses like WHERE, ORDER BY, and LIMIT to filter, sort, and limit the results.
Week 6	Practice different types of join operations
Week 7	NNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN) to combine data from multiple tables
Week 8	Create subqueries within SELECT statements to perform more complex queries
Week 9	Utilize aggregate functions (e.g., COUNT, SUM, AVG, MIN, MAX)
Week 10	Group data based on certain criteria using the GROUP BY clause. Apply the HAVING clause to filter the grouped data based on conditions.
Week 11	Practice functions like CONCAT, SUBSTRING, and LIKE
Week 12	Practice functions like CONCAT, SUBSTRING, and LIKE
Week 13	Practice writing queries with multiple levels of nested subqueries.

Week 14	Practice writing queries with multiple levels of nested subqueries.
Week 15	Implementation of an integrated database management project for each student

Learning and Teaching Resources

	Text	Available in the Library?
Required Texts	Elmasri, Ramez, and Shamkant Navathe. Fundamentals of database systems. AddisonWesley Publishing Company, 2018.	Yes
Recommended Texts	Database design, application and development.	No
Websites	http://www.sqlcourse.com/	

Grading Scheme

Group	Grade	Mark	Marks (%)	Definition
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Network Routing and switching**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2201**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	2
Administering Department	Information Technology	**College**	College of Science
Module Leader	Ali Abdulhussein Ibrahim	**e-mail**	ali.abdulhussein19@uowa.edu.iq
Module Leader's Acad. Title	Asst. Lect	**Module Leader's Qualification**	M.Sc.
Module Tutor	Ali Abdulhussein Ibrahim	**e-mail**	ali.abdulhussein19@uowa.edu.iq
Peer Reviewer Name	Dr. Maky H.Abdulraheem	**e-mail**	maky.h@uowa.edu.iq
Scientific Committee Approval Date	2025-01-20	**Version Number**	1.0

Relation with other Modules

Prerequisite module	Computer Network	**Semester**	1
Co-requisites module	-	**Semester**	-

جامعة وارث الانبياء عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents	
Module Aims	To introduce students to the fundamental concepts and principles of routing and switching in computer networks. To provide students with an in-depth understanding of network routing protocols and their role in efficient data transmission. To develop students' knowledge of network addressing and subnetting, enabling them to design and configure networks effectively. To familiarize students with switching concepts and technologies, including VLANs, spanning tree protocols, and virtualization. To equip students with the skills and techniques required to troubleshoot network connectivity and performance issues.
Module Learning Outcomes	Explain the basic principles and components of computer networks, including protocols, devices, and architectures. Understand the purpose and operation of routing protocols, such as RIP, OSPF, and BGP, and evaluate their suitability for different network environments. Design and implement IP addressing schemes and subnetting plans to efficiently allocate network resources. Configure and manage network switches, including VLANs, spanning tree protocols, and port security. Identify and resolve common network connectivity and performance issues using appropriate troubleshooting methodologies and tools.
Indicative Contents	Routing Protocols: Routing fundamentals and the role of routing protocols. Distance-vector routing protocols (e.g., RIP). Link-state routing protocols (e.g., OSPF). Border Gateway Protocol (BGP) for inter-domain routing. Network Addressing and Subnetting: IPv4 addressing and subnetting concepts. Address classes, subnet masks, and CIDR notation. Address allocation and hierarchical addressing. Switching Concepts and Technologies: Introduction to network switches and their role in local area networks (LANs). VLANs and their benefits in network segmentation. Spanning Tree Protocol (STP) and its variants. Virtual LAN Trunking Protocol (VTP) and its configuration.

Learning and Teaching Strategies	
Strategies	Lectures: In-class lectures will cover theoretical concepts, principles, and frameworks related to routing and switching. Lecturers will provide real-world examples and case studies to reinforce understanding.

	Practical Exercises: Hands-on lab sessions will allow students to apply their knowledge through network configuration and troubleshooting exercises. These practical sessions will utilize simulation software or physical network equipment. Group Discussions: Group discussions and collaborative activities will encourage students to analyze and discuss complex networking scenarios, fostering critical thinking and problem-solving skills. Online Resources: Access to online resources, including interactive tutorials, e-books, and video lectures, will supplement in-class learning and provide additional support for self-study. Assessments: Formative and summative assessments, such as quizzes, practical exams, and project assignments, will evaluate students' understanding of the concepts, their practical skills, and their ability to analyze and solve networking problems.

Student Workload (SWL)

Structured SWL (h/sem)	60	**Structured SWL (h/w)**	4
Unstructured SWL (h/sem)	87	**Unstructured SWL (h/w)**	6
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation

		Time/ Number	**Weight (Marks)**	**Weekly Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	5	10% (8)	All Weeks	1,2,3,4,5
	Assignments	5	10% (5)	All Weeks	1,2,3,4,5
	Lab	5	10% (15)	All Weeks	1,2,3,4
	Home Work	5	10% (7%)	All Weeks	1,2,3,4,5
	Projects	1	5% (5)	All Weeks	1,2,3,4,5
Summative assessment	**Midterm Exam**	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	Introduction to Network Layer
Week 2	what’s inside a router
Week 3	IP: Internet Protocol
Week 4	IP Subnetting
Week 5	DHCP and NAT
Week 6	Routing Algorithms: Link State
Week 7	Distance Vector
Week 8	Routing in the Internet: OSPF
Week 9	BGP
Week 10	Link Layer Services: Error detection, Correction
Week 11	Multiple Access Protocols
Week 12	LANs: Addressing, ARP, Ethernet and Switches
Week 13	VLANs
Week 14	Link Virtualization MPLS
Week 15	Data Center Networking
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered
Week 1	Routing Protocol Configuration and Verification
Week 2	Subnetting and IP Address Allocation
Week 3	Dynamic Routing Protocol Comparison: RIP vs. OSPF
Week 4	Static Routing Configuration and Troubleshooting
Week 5	Network Address Translation (NAT) Implementation
Week 6	Virtual LAN (VLAN) Design and Routing
Week 7	Subnet Design and Optimization for Efficient IP Addressing
Week 8	Routing Metrics and Path Selection Analysis

Week 9	Implementing VLSM (Variable Length Subnet Masking)
Week 10	IPv6 Routing Configuration and Transition Techniques
Week 11	Inter-VLAN Routing with Router-on-a-Stick Topology
Week 12	Redundancy and Load Balancing using Routing Protocols
Week 13	Routing Protocol Redistribution and Route Filtering
Week 14	Routing Loop Detection and Prevention Strategies
Week 15	Troubleshooting Routing and Subnetting Issues in a Complex Network

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	James F. Kurose and KeithW. Ross. Computer Networking: A Top-Down Approach, Eighth edition, 2020.	Yes
Recommended Texts	• L. L. Peterson and B. S. Davie. Computer Networks, A Systems Approach. Morgan Kaufman, Fourth edition, 2006. • A. S. Tanenbaum. Computer networks. Prentice-Hall, Fifth edition, 2010	No
Websites	Jim Kurose Homepage (umass.edu)	

Grading Scheme

Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	**Object-Oriented Programming II**	**Module Delivery**	
Module Type	**Core**	☒ **Lecture** ☒ **Practical**	
Module Code	**IT2202**		
ECTS Credits	**6**		
SWL (hr/sem)	**150**		
Module Level	UG2	**Semester of Delivery**	2
Administering Department	Information Technology	**College**	College Science
Module Leader	Mohsin Hassan Hussein	**e-mail**	mohsin.ha@uowa.edu.iq
Module Leader's Acad. Title	Asst. Professor	**Module Leader's Qualification**	Ph.D.
Module Tutor	Mohsen Hassan Hussein	**e-mail**	mohsin.ha@uowa.edu.iq
Peer Reviewer Name	Asst. Prof Haider Mohammed	**e-mail**	hayder.alghanami@uowa.edu.iq
Scientific Committee Approval Date	2025-01-20	**Version Number**	1.0

Relation with other Modules			
Pre-requisite module	**Object-Oriented Programming1**	**Semester**	1
Co-requisites module	-	**Semester**	

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

جامعة وارث الانبياء
عمادة كلية العلوم

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Contents

Module Aims

1. Provide a sound knowledge of the underlying principles and experience in the practical application of this course is essential for any information technology specialist.
2. extend students with procedural programming knowledge and skills in the object-oriented paradigm and builds experience with interpreted languages to introduce compiled languages.
3. In addition to further shaping a solid development methodology, the course prepares students for continued investigation into advanced programming topics.
4. develop a wide range of software solutions for real-world scenarios.

Module Learning Outcomes

On completion of this course students will be able to:

1. Compare and contrast interpreted vs compiled languages; and prototype-based vs class-based languages;
2. Competently apply the concepts of polymorphism, inheritance, encapsulation, exception handling, memory management, threads, and file I/O;
3. Design, code, verify, test, document, amend and refactor moderately complex programs meeting requirements by applying object-oriented principles;
4. Contribute to reviews of own work with others through the use of collaborative tools.
5. develop programs using the C++ Standard for real-world.

Indicative Contents

Topics

	Description	Weighting (75%)
1.	Compiled languages; imperative programming versus object orientation	5.00
2.	Objects and classes	10.00
3.	Inheritance	15.00
4.	Polymorphism	15.00
5.	Templates functions and classes	15.00
6.	Exception handling	15.00

Learning and Teaching Strategies

Strategies

Overview Strategies

Object-oriented programming is one of the principle paradigms in software development used by organisations worldwide to develop a wide range of software solutions. Sound knowledge of the underlying principles and experience in the practical application of these is essential for any information technology specialist. This intermediate programming course extends students' procedural programming knowledge and skills into the object-oriented paradigm and builds on previous experience with interpreted languages to introduce compiled languages. In addition to further shaping a solid development methodology, the course prepares students for continued investigation into advanced programming topics.

This course extends the student's basic procedural design and programming knowledge into the object-oriented paradigm. The student will be expected to learn and apply the basic concepts of object-oriented design and programming, i.e., abstraction, inheritance, and polymorphism, in the context of the C++ language through giving lectures, practical exercises within the laboratories, assignments about some specific topics, and small projects.. Key software engineering principles such as decomposition and component re-use will also be emphasised.

Student Workload (SWL)			
Structured SWL (h/sem)	75	**Structured SWL (h/w)**	5
Unstructured SWL (h/sem)	72	**Unstructured SWL (h/w)**	4.8
Student workload expectations (SWL &USWL) To do well in this subject, students are expected to commit approximately 10 hours per week including class contact hours, independent study, and all assessment tasks. If you are undertaking additional activities, the weekly workload hours may vary.			
Total SWL (h/sem)	147 + 3 final = 150		

Module Evaluation					
		Time/ Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	5	10% (8)	All Weeks	1,2,3,4,5
	Onsite Assignments	5	10% (5)	All Weeks	1,2,3,4,5
	Home Work	5	10% (7)	All Weeks	1,2,3,4,5
	Project	1	10% (5)	All Weeks	1,2,3,4,5
	Labs	5	10% (15)	All Weeks	1,2,3,4,5
Summative assessment	Midterm Exam	2hr	10% (10)	7	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)		
	Material Covered	Weighting (30+5=35%)
Week 1	- The fundamental concepts of programming, including procedural and object-oriented programming will be introduced. Also, consider the basic principles behind object-oriented programming techniques, including objects, classes, inheritance, and polymorphism. Then you will get started in programming environment by applying what you have learned.	2
Week 2	- Introduction about objects and classes, class declaration, Object declaration, with examples.	2
Week 3	**Inheritance** - Base-Class Access Control - Inheritance and protected Members - Protected Base-Class Inheritance - Inheriting Multiple Base Classes	2
Week 4	**Inheritance ...cont'd** - Constructors, Destructors, and Inheritance - When Constructor and Destructor Functions Are Executed - Passing Parameters to Base-Class Constructors	2
Week 5	**Polymorphism** - Virtual Functions - A pointer of base class type - Virtual Base Classes - Calling a Virtual Function Through a Base Class Reference	2
Week 6	**Polymorphism ...cont'd** - The Virtual Attribute Is Inherited - Virtual Functions Are Hierarchical - Pure Virtual Functions - Abstract Classes	2
Week 7	**Mid-Term Exam Revision**	2
Week 8	**Templates** - Generic Function - A Simple Function Template - A function with two generic types - What the Compiler Does	2
Week 9	**Templates ... cont'd** - Overloading a Function Template - Using Standard Parameters with Template Functions - Template Arguments Must Match	2
Week 10	**Templates ...cont'd** - Function Templates with Multiple Arguments - Template Arguments Must Match - Syntax Variation - Class Templates - An Example with Two Generic Data Types	2
Week 11	**Handling Exceptions** - Exceptions - Why Do We Need Exceptions?	2

	- Exceptions Syntax - Exception Mechanism - Throwing Exceptions - Catch Base and Derived classes with Exceptions	
Week 12	**Handling Exceptions…cont'd** - Constructor and Destructor with Exceptions - Re-thrown Exceptions - Nested Exceptions - Handling Exceptions Class Activities	2
Week 13	**Handling Exceptions…cont'd** - Handling Exceptions inside Function - Catching All Exception - Restricting Exceptions	2
Week 14	**Handling Exceptions…cont'd** - Using Threw()-To Restric any types of Exceptions - Re-thrown Exception inside function - Handling Exceptions Class Activities	2
Week 15	- Students course workload evaluation.	2
Week 16	**Prepare to the final Exam**	3

Delivery Plan (Weekly Lab. Syllabus)

	Material Covered	**Weighting (45%)**
Week 1 - Lab 1	- Prepare OOP environment, overview about unified modeling language (UML) diagram. - Access to a standard C++ or Python compiler - Linux g++ compiler and its equivalent MinGW running under windows.	3
Week 2 - Lab 2	**Introduction to Classes and Objects** - Understand function call by value method of parameter passing - Understand Pass parameters by reference method	3
Week 3 - Lab 3	**Apply Inheritance concept using many programming codes include:** - Base-Class Access Control - Inheritance and protected Members - Protected Base-Class Inheritance - Inheriting Multiple Base Classes	3
Week 4 – Lab 4	- Implement inheritance concept using Constructors, Destructors Functions and learn how to Pass Parameters to Base-Class Constructors.	3
Week 5 – Lab 5	**Implement the Polymorphism concept using many C++ code examples which include:** - Implement Virtual Functions - Use A pointer of base class type - Implement Virtual Base Classes - Calling a Virtual Function Through a Base Class Reference	3
Week 6 – Lab 6	- Implement the Virtual Attribute and Virtual Functions	3

	- Implement Pure Virtual Functions - Implement Abstract Classes	
Week 7 – Lab 7	- Labs exam1 with evaluation	3
Week 8 – Lab 8	**Implemented the Template concept using many examples code in C++:** - Understand the Generic Function - Implement A Simple Function Template - Implement function template with two generic types	3
Week 9 – Lab 9	- Implement Overloading a Function Template - Implement Using Standard Parameters with Template Functions - Prove the Template Arguments Must be Match	3
Week 10 – Lab 10	**Code examples to implement:** - Function Templates with Multiple Arguments - Class Templates - Template with Two Generic Data Types	3
Week 11 – Lab 11	**Implemented the Handling Exception concept** - Basic code exception - How can Throw Exceptions - Catch Base and Derived classes with Exceptions examples	3
Week 12 – Lab 12	**Code examples to implement:** - Constructor and Destructor with Exceptions - Re-thrown Exceptions - Nested Exceptions	3
Week 13 – Lab 13	- Labs exam2 with evaluation	3
Week 14 – Lab 14	- Code example to implement Exceptions inside Function - How to Catch All Exception? - How can Restrict Exceptions?	3
Week 15 – Lab 15	- OOP II project Implementation with discussion for each student	3

Learning and Teaching Resources		
	Text	**Available in the Library?**
Required Texts	1. Malik, D.S 2018, *C++ Programming: Program Design Including Data Structures*, 8th edn, Cengage. (ISBN 978-1-337-11756-2.) 2. OOP – Learn Object Oriented Thinking and Programming, ISBN-10: 8090466184, Tomas Bruckner, 2013. 3. The student must have access to a standard C++ compiler. The only supported compilers are the Linux g++ compiler and its equivalent MinGW running under Windows.	No

Recommended Texts	4. Object-Oriented Programming Using C++ Fourth Edition by Joyce Farrell	No
Websites		

Grading Scheme				
Group	**Grade**	Mark	**Marks (%)**	**Definition**
Success Group (50 - 100)	**A -** Excellent	Excellent	90 - 100	Outstanding Performance
	B - Very Good	Very Good	80 - 89	Above average with some errors
	C - Good	Good	70 - 79	Sound work with notable errors
	D - Satisfactory	Fair / Average	60 - 69	Fair but with major shortcomings
	E - Sufficient	Pass / Acceptable	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX –** Fail	Fail (Pending)	(45-49)	More work required but credit awarded
	F – Fail	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	Project Management	Module Delivery	
Module Type	Elective	☒ Lecture ☒ Seminar	
Module Code	IT2205		
ECTS Credits	2		
SWL (hr/sem)	50		
Module Level	UG2	Semester of Delivery	2
Administering Department	Information Technology	College	College Science
Module Leader	Elaf Adel Abbas	e-mail	Elaf.Adel.Abbas@uowa.edu.iq
Module Leader's Acad. Title	lecturer	Module Leader's Qualification	Ph.D.
Module Tutor	Lecturer Elaf Adel Abbas	e-mail	Elaf.Adel.Abbas@uowa.edu.iq
Peer Reviewer Name	Asst. lect Nabeel Sadeq	e-mail	nabeel.alshreefy@uowa.edu.iq
Scientific Committee Approval Date	2025-01-20	Version Number	1.0

Relation with other Modules

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

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

جامعة وارث الأنبياء
عمادة كلية العلوم

جامعة وارث الأنبياء
كلية العلوم
قسم تكنولوجيا المعلومات

أ.م.د شيماء حسين نوفل
٢٠٢٥/٢٠٢٤

Department Head Approval **Dean of the College Approval**

Module Aims, Learning Outcomes and Indicative Content	
Module Aims	This course aims to learn the students that the project management is the administration using the skills, tools, and methods in order to meet its requirements.
Module Learning Outcomes	1. The ability to identify the project management skills that should be used during the implementation of any project in the world of information technology. 2. The ability to analyze problems and determine the means required for a solution. 3. The ability to research and study the latest developments in the field of project management. 4. Understanding the supporting procedures for managing projects, trying to apply them, and finding the best solutions for them. 5. The ability to apply the best modern methods of project management in the field of information technology and benefit from the positive aspects and avoid the negative matters and problems that impede the implementation of the project.
Indicative Contents	Indicative content includes the following: 1. Requirements elicitation. 2. Requirements analysis. 3. Design of software, hardware, and networks. 4. System building (including software coding) and integration. 5. Verification and validation. 6. Installation. 7. Adapting the development life cycle to projects where off-the-shelf packages are to be installed.

Learning and Teaching Strategies	
Strategies	The learning and teaching strategies for studying the IT project management subject in an IT department involve: ✓ Lectures. ✓ Interactive discussions. ✓ Online resources, assessments, and feedback aid in reinforcing learning. ✓ Assessments which include individual assignments, quizzes, and examinations. These strategies ensure a comprehensive understanding of project management and its relevance in the IT field.

Student Workload (SWL)

Structured SWL (h/sem)	26	**Structured SWL (h/w)**	1.75
Unstructured SWL (h/sem)	21	**Unstructured SWL (h/w)**	1.4
Total SWL (h/sem)	47 + 3 final = 50		

Module Evaluation

		Time/Number	**Weight (Marks)**	**Week Due**	**Relevant Learning Outcome**
Formative assessment	**Quizzes**	5	10% (15)	All Weeks	1,2,3,4,5
	Onsite Assignments	3	10% (8)	All Weeks	1,2,3,4,5
	H. W	2	10% (7)	All Weeks	1,2,3,4,5
	Report	1	10% (10)	All Weeks	1,2,3,4,5
Summative assessment	**Midterm Exam**	2hr	10% (10)	5,11	
	Final Exam	3hr	50% (50)	16	
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

	Material Covered
Week 1	General Introduction of IT - Project Management. - What is a project? - What is an IT project? - Why projects?
Week 2	Project Management - What is the administration to project? - Why is project management an important skill?
	- Benefits of Project Management.

Week 3	Enterprise Project Management. - SMART Objectives. - Triple Constraint Diagram. - Effective Project Manager’s Skills.
Week 4	IT Project Characteristics. - Characteristics of A Successful Project. - Examples of IT Projects. - Project Management activities.
Week 5	- The Role of the Project Manage. - Project life cycle.
Week 6	- Purposes of Project Scheduling. - Project Management Techniques.
Week 7	Planning - Steps in planning. - WBS Concepts. - Developing the Network Plan. - Network Planning Techniques.
Week 8	PERT and CPM - Explain the Steps of PERT & CPM.
Week 9	Critical Path Method (CPM). - How to Determine the Critical Path in Project Management?
Week 10	The Work Breakdown Structure and Project Estimation. - WBS A- Definition - What is a Work Breakdown Structure?
Week 11	Managing Project Risk. - Project Risk Management. - Plan Risk Management. - Tools and Techniques. - Performance Qualitative Risk Analysis.
Week 12	Project Quality Management. - What Is Quality? - Project Quality Management Processes. - Quality Planning. - Quality Assurance plan. - Quality Control.
	- Pareto Analysis.

Week 13	Testing. - Testing Tasks in the Software Development Life Cycle. - Types of Tests - Modern Quality Management.
Week 14	The Cost of Quality. - Five Cost Categories Related to Quality. - Organization Influences, Workplace Factors, and Quality.
Week 15	Using Software to Assist in Project Quality Management.
Week 16	**Preparatory week before the Final Exam**

Learning and Teaching Resources

	Text	**Available in the Library?**
Required Texts	An Introduction to Project Management, Sixth Edition, by Kathy Schwalbe	
Recommended Texts		
Websites	- https://www.projectmanager.com/guides/it-project-management - https://www.techtarget.com/searchcio/definition/IT-project-management	

Grading Scheme

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