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	**Ali Mahmoud Ali**		**e-mail**	ali.mahmoud@uowa.edu.iq	
Module Leader's Acad. Title	Asst. Lecturer		**Module Leader's Qualification**	MSc	
Module Tutor	**Ali Mahmoud Ali**		**e-mail**	ali.mahmoud@uowa.edu.iq	
Peer Reviewer Name	Dr. Haider M.Ali		**e-mail**	hayder.alghananmi@uowa.edu.iq	
Scientific Committee Approval Date	2025-09-1		**Version Number**	V1.0	

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

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	**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	Elaf Ali Sfooq	**e-mail**	elaf.safooq@uowa.edu.iq	
Module Leader's Acad. Title	Lecturer	**Module Leader's Qualification**	MSc	
Module Tutor	Elaf Ali Sfooq	**e-mail**	elaf.safooq@uowa.edu.iq	
Peer Reviewer Name	Dr.Maky H.Abdulraheem	**e-mail**	maky.h@uowa.edu.iq	
Scientific Committee Approval Date	2025-09-1	**Version Number**	V1.0	

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

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	**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	Maky H.Abdulraheem		**e-mail**	maky.h@uowa.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	Ph.D	
Module Tutor	Ali Abdul Hussein Ibrahim		**e-mail**	ali.abdulhussein@uowa.edu.iq	
Peer Reviewer Name	Dr .Hayder Ghanim		**e-mail**	hayder.alghanami@uowa.edu.iq	
Scientific Committee Approval Date	2025-09-1		**Version Number**	V01	

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

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	**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**	Msc	
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	2025-09-1		**Version Number**	V1.0	

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

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	**Object-oriented programming I**			**Module Delivery**	
Module Type	**Core**			**☒Lecture**	
Module Code	**IT2112**			**☒Practical**	
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	2025-09-17	**Version Number**	V1.0		

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

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

<table>
<tr><th colspan="2">Module Aims, Learning Outcomes and Indicative Contents</th></tr>
<tr><td>Module Aims</td><td>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.</td></tr>
<tr><td>Module Learning Outcomes</td><td>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.</td></tr>
<tr><td>Indicative Contents</td><td><u>Topics</u>
<u>] scription</u> <u>Weighting (75%)</u>
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</td></tr>
</table>

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	**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 Mohsin**	**e-mail**	karar.sadeq@uowa.edu.iq
Module Leader's Acad. Title	Asst. Lecturer	**Module Leader's Qualification**	MS.c
Module Tutor	Ali Abdul Hussein 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	2025-09-1	**Version Number**	V01

Relation with other Modules

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

جامعة وارث الانبياء
كلية العلوم
قسم تكنولوجيا المعلومات

جامعة وارث الانبياء
عمادة كلية العلوم

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.