MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Distributed Computing & Systems**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☐ **Lecture** ☒ **Lab** ☒ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**CET4108**				
ECTS Credits	**5.00**				
SWL (hr/sem)	**150**				
Module Level	3		**Semester of Delivery**	5	
Administering Department			**College**		
Module Leader	Dr. Mohanad Ahmed		**e-mail**	Mohanad.ahmed1@mtu.edu.iq	
Module Leader's Acad. Title	Lecturer		**Module Leader's Qualification**	Ph.D.	
Module Tutor	Asst Prof. Siraj Qays Mahdi		**e-mail**	Siraj_qays@mtu.edu.iq	
Peer Reviewer Name	Asst. Prof. Alhamzah Taher Mohammed		**e-mail**	alhamza_tm@mtu.edu.iq	
Scientific Committee Approval Date			**Version Number**	1.0	

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

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	1. To introduce students to the fundamental problems, concepts, and approaches in the design and analysis of distributed computing systems. 2. To familiarize students with the stages of the distributed system design cycle, including system architecture, data and processes arrangements, naming, communication and coordination issues, existing distributed computing paradigms, techniques, and tools, and evaluating the effectiveness of distributed application systems for specific data, task, and user types. 3. understand the evolution of the distributed computing from its early beginnings as multi-processor and multi-computer systems, to computer networks, to the emerging cloud, edge (fog, dew, mist) and heterogeneous computing environments 4. explain and discuss basic principles and typical examples of real-world distributed systems such as NFS file-sharing system and the web. 5. understand process migration or more specifically code migration and its role in achieving scalability of distributed system
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	1. State, prove and apply Shannon's sampling theorem know the widely used models of communication: Remote Procedure Call (RPC), and Message-Oriented Middleware (MOM). 2. know basic principles of the RPC model and problems with achieving distribution transparency. 3. understand the peculiarities of the high-level message-queuing model of process communication. 4. know how to set up multicast facilities for data dissemination in distributed systems. 5. understand traditional deterministic means of multicasting as well as probabilistic approaches. 6. know the use of Domain Name System (DNS). 7. know the way of using attributes assigned to an entity to resolve a description of an entity in distributed system. 8. know basics of the security management including mechanisms to distribute cryptographic keys, add and remove users from a system, prove ownership to access specified resources, etc...
Indicative Contents المحتويات الإرشادية	Indicative content includes the following.

Introduction: Design goals

Distributed systems consist of autonomous computers that work together to give appearance of a single coherent system. Design goals for distributed systems include sharing resources and ensuring openness. In addition, designers aim at hiding many of the intricacies related to distribution of processes, data and control. [10 hrs]

Types of systems

Different types of distributed systems exist which can be classified as being oriented towards supporting computations, information processing and pervasiveness. Distributed computing systems are typically deployed for high-performance applications often originating from parallel computing. Cloud computing goes beyond high-performance computing and also supports distributed systems found in traditional office environments. An emerging class of distributed systems is represented by pervasive computing environments, including mobile-computing systems as well as sensor-reach environments. [10 hrs]

Architectures: Architectural styles; Middleware

We can make a distinction between software architecture and system architecture. AN architectural style reflects the basic principle that is followed in organizing the interaction between the software components comprising a distributed system. Important styles include layering, object-based styles, resource-based styles, and styles in which handling events are prominent. [10 hrs]

Communication: Foundations. RPC

Communication between processes is essential for any distributed system. In traditional network applications, communication is often based on the low-level message-passing primitives offered by the transport layer. One of the most widely used abstractions is the Remote Procedure Call (RPC), that offers synchronous communication facilities, by which a client is blocked until the server has sent a reply. [10 hrs]

Communication: Message-oriented & Multicast communication

Message-oriented middleware models generally offer persistent asynchronous communication, and are used where RPCs are not appropriate. An important class of communication protocols in distributed systems is multicasting. [10 hrs]

Naming: Names, IDs. Flat naming

Names are used to refer to entities. There are three types of names: an address, an identifier, and human-friendly names. Given these types, we make a distinction

	between flat naming, structured naming, and attribute-based naming. Systems for flat naming essentially need to resolve an identifier to the address of its associated entity. This can be done in different ways. [10 hrs] Fault tolerance Fault tolerance is defined as the characteristic by which distributed computing system can mask the occurrence and recovery from failures. Several types of failures exist. Redundancy is the key technique needed to achieve fault tolerance. When applied to processes, the notion of process groups becomes important. The real problem is that members of a process group need to reach consensus in the presence of various failures. Paxos is by now a well-established and highly robust consensus algorithm. [10 hrs]

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The assessment strategy for this module is designed to provide students with the opportunity to demonstrate the skills and knowledge as described in the learning outcomes. The written examination will assess the knowledge of terminology, concepts and theory of Distributed Computing & Systems, as well as the ability to analyze problems and new features of distributed computing systems to solve and predict effects. The laboratory experiments will evaluate the acquired technical skills and expertise required to apply these methods.

Student Workload (SWL) الحمل الدراسي للطالب			
Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	74	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	5.3
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	76	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	5.4
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	150		

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

Delivery Plan (Weekly Syllabus) المنهاج الاسبوعي النظري	
	Material Covered
Week 1	**Design goals** Distributed systems consist of autonomous computers that work together to give appearance of a single coherent system.
Week 2	**Architectures: System architecture. Example** There are many different organizations of distributed systems. Client-server architectures are often highly centralized.
Week 3	**Processes: Threads. Virtualization** Processes play a fundamental role in distributed systems as they form a basis for communication between different machines.
Week 4	Processes: Clients. Servers.
Week 5	**Communication: Foundations. RPC**
Week 6	**Naming: Structured naming. Attribute-based naming**
Week 7	Mid-term Exam
Week 8	**Coordination: Clock synchronization** There are various ways to synchronize clocks in a distributed system. All methods are based on exchanging clock values, while taking into account the time it takes to send and receive messages.
Week 9	**Coordination: Mutual exclusion. Election algorithms**

	An important class of synchronization algorithms is that of distributed mutual exclusion. These algorithms ensure that in a distributed collection of processes, at most one process at a time has access to a shared resource.
Week 10	Consistency and replication: Data-centric & Client-centric models
Week 11	Consistency and replication: Replica management. Consistency protocols
Week 12	Fault tolerance.
Week 13	Security
Week 14	**Distributed System Models and Enabling Technologies**
Week 15	**Parallel Programming Systems and Models**
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Introduction to Distributed Systems. (Build a Cluster)
Week 2	Lab 2: Distributed System Models and Enabling Technologies. (Benchmarking)
Week 3	Lab 3: Parallel Programming Systems and Models. (Map Reduce)
Week 4	Lab 4: Cloud Computing.
Week 5	Lab 5: File systems & Networked file systems.
Week 6	Lab 6: Optimizing Large Data Transfers over 100Gbps Wide Area Networks - Anupam Rajendran.
Week 7	Lab 7: Distributed Filesystems.

Learning and Teaching Resources مصادر التعلم والتدريس		
	Text	**Available in the Library?**
Required Texts	Tanenbaum S. Maarten V.S.: Distributed Systems Principles and Paradigms, (Pearson Education).	No
Recommended Texts	George Coulouris, Jean Dollimore. Tim Kindberg: Distributed Systems concepts and design.	No
Websites	https://www.youtube.com/watch?v=cQP8WApzIQQ&list=PLrw6a1wE39_tb2fErI4-WkMbsvGQk9_UB&ab_channel=MIT6.824%3ADistributedSystems	

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

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