Course unit title:	Fundamentals of Distributed Systems
Course unit code:	CSC330
Type of course unit: **(Compulsory/optional)**	Compulsory
Level of course unit: **(First, second or third cycle)**	Bachelor (1st cycle)
Year of study:	3
Semester when the unit is delivered:	6
Number of ECTS credits allocated:	6
Name of lecturer(s):	TBA

Learning outcomes of the course unit:

Upon succesful completion of this course students should be able to:

- Explain and discuss the principles and theoretical models used in designing distributed systems.
- Describe the trade-offs which must be made when designing a distributed system.
- Describe and evaluate algorithms and architectural models used in implementing distributed file systems, logical clocks, elections, mutual exclusion, multicast message ordering, transactions, replication and peer-to-peer networks in distributed systems.
- Apply the algorithms and models taught to implement programs that perform distributed computation, through the application of taught protocols.
- Evaluate the suitability of algorithms, methods and models for a given application.

Mode of delivery:	Face- to- face
Prerequisites and co-requisites:	CSC322 or CSW241
Recommended optional program components:	None

Course Contents:

Objective:
This course studies the key design principles of distributed systems, which are collections of independent networked computers that function as single coherent systems. It covers fundamental concepts of distributed systems including network architectures, communication protocols, processes and threads and naming. It covers important paradigms in distributed systems, including logical clocks, distributed mutual exclusion; consistency, replication, fault

tolerance, coordination and agreement and security. It addresses failures and fault-tolerance techniques in diverse applications, such as consensus, transactions, replicated data management, and self-stabilization.

Description:
Fundamentals:
definition of a distributed system, properties of distributed systems (distribution transparency, openness), scalability, types of distributed systems. architectures of distributed systems. processes, threads, virtualization, clients, servers, code migration.

Communication:
layered protocols, types of communication, remote procedure call, message-oriented communication, stream-oriented communication, multicast communication.

Naming:
names, identifiers, and addresses, flat and structured naming, attribute-based naming.

Synchronization:
clock synchronization, physical clocks, global positioning system, clock synchronization algorithms, logical clocks, lamport's logical clocks, vector clocks. mutual exclusion: centralized, decentralized, distributed algorithm, a token ring algorithms, comparison of them. Election algorithms: traditional election algorithms, elections in wireless environments, elections in large-scale systems.

Consistency and replication:
reasons for replication, data-centric consistency models, client-centric consistency models: eventual consistency, monotonic reads & writes. Replica management, consistency protocols.

Fault tolerance:
basic concepts, failure models, process resilience: failure masking and replication, agreement in faulty systems, failure detection. reliable client-server communication: point-to-point communication, reliable group communication: basic reliable-multicasting schemes, scalability in reliable multicasting, atomic multicast. Distributed commit, recovery.

Security: introduction, secure channels, access control.
Distributed object-based systems, distributed file systems, distributed web-based systems, distributed coordination-based systems: architecture, processes, communication, naming, synchronization, consistency and replication, fault tolerance.

Recommended or required reading:	A. Tannenbaum, M. van Steen, DISTRIBUTED SYSTEMS: PRINCIPLES AND PARADIGMS, Prentice Hall (2nd edition) 2006. G. Coulouris, J. Dollimore, T. Kindberg, DISTRIBUTED SYSTEMS: CONCEPTS AND DESIGN, 4th edition, Addison-Wesley, 2005

	Sukumar Ghosh, DISTRIBUTED SYSTEMS: AN ALGORITHMIC APPROACH, 2006 CRC Press H. Attiya and J. Welch, DISTRIBUTED COMPUTING, FUNDAMENTALS, SIMULATIONS, AND ADVANCED TOPICS, Second Edition, John Wiley and Sons, Inc. Gerard Tel, INTRODUCTION TO DISTRIBUTED ALGORITHMS, Cambridge University Press 2000.
Planned learning activities and teaching methods:	Class Instruction 42 Hours Consultation/Computer Lab 30 Hours
Assessment methods and criteria:	Examinations 60% Assignments 40% 100%
Language of instruction:	English
Work placement(s):	No
Place of Teaching:	Theoretical Part: Regular Classroom European University Cyprus, Nicosia Practical Part: Computer Laboratory European University Cyprus, Nicosia