Course Title	Operating Systems				
Course Code	ACSC271				
Course Type	Compulsory				
Level	BSc (Level 1)				
Year / Semester	2nd (Fall)				
Teacher's Name	Dr Christos Markides				
ECTS	5	Lectures / week	3	Laboratories/week	N/A
Course Purpose	The course introduces Computer Science and Computer Engineering students to the concepts of Operating Systems. The course aims to outline the principles and the underlying mechanisms of modern Operating Systems in understanding how an operating system allows the interaction between machine and user and the allocation of resources. The course considers the concepts of modern operating systems. Teaches abstracts of processes, CPU scheduling, memory management, file systems, and security. Implements applications based on UNIX (commands, syntax, programming). Discusses alternative operating systems, such Ubuntu and CentOS, Linux, Windows 7, 8, and 10 with practical exercise on Linux OS and Windows.				
Learning Outcomes	By the end of the course, the students are expected to: 1. Describe computer organisation and architecture, as well as the structure and operations of an operating system with regards to resource management (CPU, memory and storage) and explain the concepts of CPU scheduling, process synchronisation and virtual memory. 2. Explain scheduling algorithms to solve synchronisation problems for processes and threads and memory usage (main and virtual) with the use of system calls for single and multiprocessor systems. 3. Examine the critical-section problem in process synchronisation and outline the use of semaphores for classic problems of synchronisation. 4. Evaluate and compare resource management, file-system interface, protection and security of different operating systems. 5. Design and construct programs for monitoring system resources and applying scheduling algorithms for process management and synchronisation.				
Prerequisites	ACSC110, ACSC182	Co-requisites	None		
Course Content	• **Overview**: Operating System definition and services, Common types of Operating Systems, interrupt driven operating systems, System Components Protection and Security, System Calls and Types of System Calls. • **Processes**: Definitions, Process States, diagrams, Parent, Child,				

	memory sharing and the PCB, Context Switching and diagram, Process Scheduling, Process Creation, Windows and Linux Processes. • **Threads**: Definitions, Single and Multithreaded, Multithreading Programming, Thread Context Switching, Benefits, User and Kernel Threads, Multithreading Models and Diagrams, Threading Issues, Windows and Linux Threads. • **CPU Scheduling**: Overview, CPU Scheduler and Dispatcher, Scheduling criteria, Optimisation criteria, Scheduling Algorithms, Gantt Charts, Pre-emptive and Non-preemptive Scheduling, and Algorithm Evaluation • **Process Synchronisation**: Producer / Consumer, Solution to Critical Section Problem, Algorithms and Semaphores, Deadlock and Starvation, Synchronisation Problems – Dining Philosophers Problem. • **Memory**: Overview, Binding Instructions and Data to Memory, Memory Management, Memory Management Unit, Swapping, Dynamic Loading, and Dynamic Linking Contiguous Allocation, Dynamic Storage Allocation, Fragmentation, Page faults sand segmentation. • **Virtual Memory**: Overview and Implementation, Demand and pure demand paging, Page Tables and Page Faults, Page Replacement and Algorithms, Page Fault and Page fault Correction. • **File System Interface**: File system, Definitions and operations on files and directories, Different structures (NAS, SAN), File systems, Different storage methods and Scheduling Algorithms.
Teaching Methodology	The taught part of course is delivered to the students by means of lectures, conducted with the help of computer presentations. Lecture notes and presentations are available through the e-learning platform and the web for students to use in combination with the textbooks. The course will combine theoretical aspects of operating systems with some practical work with the concepts of operating systems. Delivery will be based on 2 period lecturing and 1 laboratory period. Laboratory work will mainly consist of introducing students to the practical and students are expected to complete the practical outside contact hours.
Bibliography	Textbooks: • Abraham Silberschatz, Peter Baer Galvin, "***Operating System Concepts***", 10th Edition, Wiley, 2018, ISBN: 978-1119456339. References: • A. S. Tanenbaum, "***Modern Operating Systems***", 4th Edition, Pearson, 2016, ISBN: 978-9332575776. • W. Stallings, " ***Operating Systems: Internals and Design Principles***", 9th Edition, Pearson, 2017, ISBN: 978-0134670959.
Assessment	Students are assessed on the theoretical aspects of the course through tests, and the final exam, while practicals cover the applied and hand-on aspects of the course. Coursework will comprise of two tests, a set of practicals, one assignment, and three-hour closed book exam. The weights for each assessment component are: • Assignment: 10%

	• Practicals: 15% • Tests: 25% • Final Exam: 60%
Language	English