Title (Units): **COMP2017 Operating Systems (3,3,1)**

Course Aims: To introduce the fundamentals and major concepts of operating systems design and implementation; to study the detailed operations of various components of an operating system.

Prerequisite: (i) COMP2006 Computer Organization
(ii) COMP2045 Programming and Problem Solving OR COMP2046 Problem Solving Using Object Oriented Approach OR COMP2056 Object-Oriented Programming and Java

Course Intended Learning Outcomes (CILOs):
Upon successful completion of this course, students should be able to:

No.	Course Intended Learning Outcomes (CILOs)
	Knowledge
1	Describe the basic concepts of operating systems, including development and achievements, functionalities and objectives, structure and components
2	Explain how memory, I/O devices, files, processes and threads are managed, and evaluate the performance of various scheduling algorithms
3	Explain the concepts covered in concurrency control, including mutual exclusion and synchronization, deadlock and starvation
4	Develop software using multiprocess and multithread programming techniques
	Professional Skill
5	Analyze the relationship between the operating system and the hardware environment in which it runs
	Attitude
6	Develop the role of operating systems in a wider context, e.g., extending OS services via system calls

Calendar Description: This course introduces the fundamentals of operating systems design and implementation. Topics include an overview of the components of an operating system, mutual exclusion and synchronization, deadlock and starvation, implementation of processes and threads, resources scheduling algorithms, memory management, and file systems.

Teaching and Learning Activities (TLAs):

CILOs	Type of TLA
1-3, 5	Students will learn operating systems concepts via lectures, tutorials and continuous assessment activities.
3-4, 6	Students will acquire hands-on experience in multiprocess / multithread programming, process synchronization and deadlock, and extending OS services via lectures, laboratory exercises and continuous assessment activities.

Assessment:

No.	Assessment Methods	Weighting	CILOs to be addressed	Description of Assessment Tasks
1	Continuous Assessment	40%	1-6	Continuous assessment activities are designed to assess students' mastery of the major concepts of operating systems and the associated hands-on skills. These activities include both written work (e.g., written assignment or test) and hands-on work (e.g., programming assignment or practical test).
2	Examination	60%	1-5	Final examination questions are designed to assess how far students have achieved their intended learning outcomes. Questions will primarily be analysis and skills based to assess students' ability in operating systems design and implementation.

Assessment Rubrics:

Criteria	Excellent (A)	Good (B)	Satisfactory (C)	Marginal Pass (D)	Fail (F)
Basic objectives, functionalities, and components in an operating system	The student acquires excellent knowledge in the objectives of operating systems, how operating systems are related to computer hardware, what functionalities are provided to users, and what the major components are in operating systems.	The student acquires sufficient knowledge in the objectives of operating systems, how operating systems are related to computer hardware, what functionalities are provided to users, and what the major components are in operating systems.	The student acquires average knowledge in the objectives of operating systems, how operating systems are related to computer hardware, what functionalities are provided to users, and what the major components are in operating systems.	The student is able to identify major components in an operating system, name some basic functionalities in an operating system, and briefly describe how these components or functionalities work.	The student is unable to identify major components in an operating system, name some basic functionalities in an operating system, and briefly describe how these components or functionalities work.
Processes and Threads	The student understands thoroughly the internal structures of processes and threads, what mutual exclusion is, how to synchronize processes and avoid deadlocks, and how to schedule processes and threads.	The student understands sufficiently the internal structures of processes and threads, what mutual exclusion is, how to synchronize processes and avoid deadlocks, and how to schedule processes and threads.	The student understands more than basic concepts in the internal structures of processes and threads, what mutual exclusion is, how to synchronize processes and avoid deadlocks, and how to schedule processes and threads.	The student understands some basic concepts of processes and threads, deadlocks, and process/thread scheduling algorithms.	The student has no knowledge in basic concepts of processes and threads, deadlocks, and process/thread scheduling algorithms.
Memory, I/O devices, and files	The student demonstrates excellent know-how in the management of main and virtual memory, I/O devices, and files.	The student demonstrates considerable know-how in the management of main and virtual memory, I/O devices, and files.	The student demonstrates average know-how in the management of main and virtual memory, I/O devices, and files.	The student demonstrates some know-how in the management of main and virtual memory, I/O devices, and files.	The student does not know about the management of main and virtual memory, I/O devices, and files.
Multiprocess and/or multithread programming techniques	The student correctly writes multiprocess/multithread programs that meet all specifications.	The student writes multiprocess/multithread programs that meet most specifications.	The student writes multiprocess/multithread programs that meet some key specifications.	The student writes multiprocess/multithread programs that meet a few specifications.	The student is unable to program with multiprocess/multithread techniques.

Course Content and CILOs Mapping:

Content		**CILO No.**
I	Operating Systems Overview	1, 5-6
II	Process & Thread Management	2-6
III	Concurrency Control	3-4, 6
IV	Memory Management	2, 5-6
V	Processor Scheduling	2, 5-6

VI	I/O & File Management	2, 5-6
VII	Case Studies	1-6

References:

- A. Silberschatz, P.B. Galvin, G. Gagne, Operating System Concepts, Addison-Wesley, 10th Edition. 2019.
- W. Stallings, Operating Systems: Internals and Design Principles, Prentice-Hall, 9th Edition. 2017.
- A. S. Tanenbaum and H. Bos, Modern Operating Systems, 5th Edition. Pearson, 2023.

Course Content:

Topic

I. Operating Systems Overview
   1. Historical development
   2. Operating system objectives and functionalities
   3. Major achievements

II. Process & Thread Management
   1. Process concepts
   2. Thread concepts
   3. Descriptions, structures, and controls
   4. Multiprocess and Multithread programming

III. Concurrency Control
   1. Mutual exclusion
   2. Synchronization
   3. Deadlock
   4. Starvation

IV. Memory Management
   1. Multiprogramming and partitions
   2. Paging and segmentation
   3. Virtual memory
   4. Demand paging
   5. Page replacement algorithms

V. Processor Scheduling
   1. Scheduling concepts
   2. Uniprocessor and multiprocessor scheduling
   3. Algorithm evaluation

VI. I/O & File Management
   1. I/O devices
   2. Disk scheduling
   3. File organization
   4. Directory structures

VII. Case Studies