Title (Units): **COMP 2320 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: Nil

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

No.	Learning Outcomes (LOs)
	Knowledge
1	Describe the basic concepts of operating systems, including development and achievements, functionalities and objectives, structure and components
2	Explain how memory, devices (I/O), files, processes and threads are managed, and evaluate the performance of various scheduling algorithms
3	Define 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
6	Design and implement appropriate operating systems
	Transferable Skill
7	Solve complex problems in groups and develop skills in problem solving using systematic approaches
	Attitude
8	Commit to develop the role of operating systems in wider context

Calendar Description: 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.

Assessment:

No.	Assessment Methods	Weighting	Remarks
1	Continuous Assessment	30%	Continuous assessments are designed to measure how well students have learned the fundamentals and major concepts of operating systems.
2	Examination	70%	Final examination questions are designed to see 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.

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,	The student acquires **sufficient** knowledge in the objectives of operating systems, how operating systems are related to computer hardware, what functionalities are	The student acquires **average** knowledge in the objectives of operating systems, how operating systems are related to computer hardware, what functionalities	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	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

	and what the major components are in operating systems.	provided to users, and what the major components are in operating systems.	are provided to users, and what the major components are in operating systems.	functionalities work.	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.

Learning Outcomes and Weighting:

Content	**LO No.**
I. Operating Systems Overview	1, 5-8
II. Process & Thread Management	2, 4-8
III. Concurrency Control	3-8
IV. Memory Management	2, 5-8
V. Processor Scheduling	2, 5-8
VI. I/O & File Management	2, 5-8
VII. Case Studies	1-8

References:

H.M. Deitel, P.J. Deitel, D.R. Choffnes, Operating Systems, 3rd Edition., Addison-Wesley, 2004.
A. Silberschatz, P.B. Galvin, Operating System Concepts, Addison-Wesley, 7th Edition. 2004.
W. Stallings, Operating Systems, Prentice-Hall, 5th Edition. 2004.
A.S. Tanenbaum, Modern Operating Systems, 3rd Edition. Prentice-Hall, 2007.

Course Content in Outline:

Topic

I. Operating Systems Overview
   A. Historical development
   B. Operating system objectives and functionalities
   C. Major achievements

II. Process & Thread Management
   A. Process concepts
   B. Thread concepts
   C. Descriptions, structures, and controls
   D. Multiprocess and Multithread programming

III. Concurrency Control
   A. Mutual exclusion
   B. Synchronization
   C. Deadlock
   D. Starvation

IV. Memory Management
   A. Multiprogramming and partitions
   B. Paging and segmentation
   C. Virtual memory
   D. Demand paging
   E. Page replacement algorithms

V. Processor Scheduling
   A. Scheduling concepts
   B. Scheduling algorithms
   C. Algorithm evaluation

VI. I/O & File Management
   A. I/O devices
   B. Disk scheduling
   C. File organization
   D. Directory structures

VII. Case Studies