COURSE DESCRIPTION FORM

INSTITUTION Hamdard University

PROGRAM (S) TO BE EVALUATED BS (Computer Science)

A. Course Description

Course Code	CS 223
Course Title	Operating Systems
Credit Hours	3 + 1
Prerequisites by Course(s) and Topics	CS111: Introduction to Computing
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Assignments / Quizzes 20 Marks Midterm Examination 30 Marks Final Examination 50 Marks Lab 50 Marks
Course Coordinator	Iqbal Uddin Khan
URL (if any)	
Current Catalog Description	This course intended as a general introduction to the techniques used to implement operating systems and related kinds of systems software. Among the topics covered will be process management (creation, synchronization, and communication); processor scheduling; deadlock prevention, avoidance, and recovery; main-memory management; virtual memory management (swapping, paging, segmentation and page-replacement algorithms); control of disks and other input/output devices; file-system structure and implementation; and protection and security. Most importantly, the interactions between these concepts are examined.
Textbook (or **Laboratory Manual** for Laboratory Courses)	Operating Systems Concepts, 9th Edition by Silberchatz and Galvin, ISBN-10: 1118063333, Publisher: WILEY
Reference Material	Operating Systems, Internal and Design Principles, 8th Edition by William Stallings, ISBN-10: 0133805913, Publisher: PEARSON Modern Operating Systems, 3rd Edition by Andrew S. Tanenbaum

Course Goals	This course has two components: a theory component to teach you the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation. In the theory component, you will learn about processes and processor management, concurrency and synchronization, memory management schemes, file system and secondary storage management, security and protection, etc. At the end of the course you should: • Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc., • Understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions, • Understand how the operating system abstractions can be implemented, • Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software, • Understand basic resource management techniques (scheduling or time management, space management) and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.
Topics Covered in the Course, with Number of Lectures on Each Topic (assume 15-week instruction and one-hour lectures)	**1. Operating System Introduction (3)** a. Computer System Organization and Architecture b. Operating System Structure and Operations c. Computing Environments **2. Operating System Structures (3)** a. Operating System Services b. User and Operating System Interface c. System Calls, System Programs and System Boot **3. Processes (3)** a. Process Concept, Control Block, Scheduling b. Operations over Processes c. Inter Process Communication **4. Threads (3)** a. Concept of Threads b. Multi-Core Programming c. Multithreading Models **5. Threads (Continued) (3)** a. Thread Libraries b. POSIX Thread Libraries **6. Process Synchronization (3)** a. Concept of Synchronization b. Critical Section Problem c. Scheduling Algorithms **7. Process Synchronization (Continued) (3)** a. Peterson's Solution b. Semaphores c. Monitors **8. Scheduling (3)** a. Concept of Scheduling

	b. Scheduling Algorithms c. Examples and Gant Charts for Scheduling Algorithms d. Thread Scheduling **9. Deadlock (3)** a. System Model b. Deadlock Characterization c. Deadlock Prevention **10. Deadlock (Continued) (3)** a. Deadlock Avoidance b. Deadlock Detection c. Recovering from Deadlock **11. Main Memory (3)** a. Address Binding and Address Spaces b. Swapping, Contiguous Memory Allocation c. Segmentation, Paging, Structure of Page Table **12. Virtual Memory (3)** a. Introduction of Virtual memory b. Demand Paging c. Copy on Write **13. Virtual Memory (Continued) (3)** a. Page Replacement b. Allocation of Frames c. Thrashing **14. Mass Storage (3)** a. Disk Structure – Attachment – Scheduling b. Swap Space Management c. RAID Structure **15. File System (3)** a. File Concepts b. Directory and Disk Structure c. File System Mounting, Sharing and Protection d. File System Structure
Laboratory Projects/Experiments Done in the Course	• Introduction to Operating Systems • Bootloader • Introduction to Command Line Interface • System Calls (Linux and Windows) • Adding Modules in Kernel • Inter - Processes Communication • Race Condition and Zombie Processes • Process Synchronization • Windows and Linux Thread Libraries • Scheduling Schemes • Deadlocks • Memory Management • Virtual Memory • File System Access and Control Systems • Disk Management Algorithms
Programming Assignments Done in the Course	Please refer to lab manual for details.

Class Time Spent on (in credit hours)	**Theory**	**Problem Analysis**	**Solution Design**	**Social and Ethical Issues**
	12	18	10	08
Oral and Written Communications	Every student is required to submit at least 04 written reports of typically 3 - 6 pages and to make 02 oral presentations of typically 3 - 5 minute's duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.			

Instructor Name: **_ Iqbal Uddin Khan_____**

Instructor Signature: _____

Date: _____