Far Western University
Faculty of Education
Bachelor in Computer Science Education

Course Title: Discrete Structure	Nature: Theory and Practical
Course No.: CS.Ed.233	Credits: 3
Level: B.Ed.	Total Periods: 48+16
Third Semester	Practical: 2 hours per period

1. Course Description

This course is developed to introduce the fundamental concept of discrete mathematics and how to apply mathematical concept in computer science. This course provide the way that how to represent logics in real world and it also provide the knowledge of mathematical reasoning, algorithmic thinking and application modeling.

2. Course Objectives

The objective of this course is to teach the fundamental concepts of discrete mathematics that will enable students to solve the interesting and challenging real world problems. This course does not directly help us write programs. At the same time, it is the mathematics underlying of almost all of computer science. At the end of this course the students should be able to:

- Gain fundamental and conceptual clarity in the area of logic, reasoning, algorithms, recurrence relation, and graph theory.
- Design high-speed networks and message routing paths.
- Find good algorithms for sorting.
- Design the good web search algorithm.
- Analyze algorithms for correctness and efficiency.
- Formalizing security requirements.
- Designing cryptographic protocols

3. Specific objectives and Course Contents:

Specific objectives	Unit 1: Logic, Induction and Reasoning (13 hrs.)
• Introduce the concept of propositional logic • Implement the expression in propositional logic • Describe the concept of predicate logic • Introduce the concept of inference • Explain the rules of inference • Describe the inference rules for quantified statement • Illustrate different proofing techniques • Differentiate between induction and complete induction	1.1. Proposition and Truth table and connectives 1.2. Propositional Logic 1.3. Expressing statements in Propositional Logic 1.4. The predicate Logic and quantifiers 1.5. Negation of quantified statements 1.6. Nested quantifiers 1.7. Rules of Inference and inference rules for quantified statement. 1.8. Proof Methods (Direct Proof, Indirect Proof, Proof by Contradiction, Proof By Contraposition, Exhaustive Proofs and Proof by Cases), 1.9. Mathematical Induction and strong Induction

	Unit 2: Integers and Relations (10hrs.)
• Introduce the concept of integer, prime number and GCD • Describe Extended Euclidean algorithm • Illustrate the concept of application of number theory • Describe Chinese Remainder Theorem • Explain the concept of relation and their representation.	2.1. Integers: Integers and Division, Division Algorithm, 2.2 Modular Arithmetic, Primes and Greatest Common Divisor, 2.3 Extended Euclidean Algorithm, Integers and Algorithms(Addition, Multiplication, Division & Remainder Algorithms) 2.4 Applications of Number Theory (Linear Congruencies, Chinese Remainder Theorem, Computer Arithmetic with Large Integers. 2.5 Relations: Relations and their Properties, Combining Relations, N-ary Relations, Operations on N-ary Relations, 2.6 Applications of N-ary Relations, Representing Relations by using Matrix and Diagraphs
	Unit 3. Counting and Discrete Probability (11 hrs.)
• Describe counting, sum and product rule. • Explain the use of pigeonhole principle and generalized pigeonhole principle. • Illustrate the concept of permutation and combination. • Describe the use of binomial theorem. • Explain the concept of probability, discrete probability and conditional probability. • Explain the concept of recurrence relation and solution of recurrence relation.	3.1 Counting: Basics of Counting, Sum and Product Rule, Counting Problems. 3.2 Pigeonhole Principle, Generalized Pigeonhole Principle. 3.3 Applications of Pigeonhole Principle. 3.4 Permutations and Combinations, Two Element Subsets, Counting Subsets of a Set. 3.5 Binomial Coefficients, Binomial Theorem (without proof). 3.6 Pascal's Identity and Triangle. 3.7 Discrete Probability: Introduction to Discrete Probability, Probability Theory, Conditional Probability, Independence, random Variable, Probabilistic Primility Testing, Expected Value and Variance. 3.8. Recursive Definition of Sequences 3.9. Solution of Linear recurrence relations (statement of theorem 1 to theorem 4 without proof). 3.10. Solution to Nonlinear Recurrence Relations (statement of theorem 5 and theorem 6 without proof) 3.11. Application to Algorithm Analysis
	Unit 4. Graph Theory (14 hrs.)
• Introduce the concept of graph • Explain the different way of representing graph • Explore the basic concept of path and circuits • Working on connectedness algorithms • Find the shortest path between source and destination • Deduce the concept of planarity testing	4.1. Graph introduction, types of graph (Undirected and Directed Graphs), simple graph, multigraph, pesudograph and graph representation (adjacency list, adjacency matrix and incidence matrix for directed and undirected graph) 4.2 Degree of graph (directed and undirected) and Handshaking theorem 4.3. Walk Paths, Circuits, Components 4.4. Connectedness Algorithm 4.5. Shortest Path Algorithm 4.6. Bipartite Graphs, Planar Graphs, Regular Graphs

• Differentiate between Eulerian and Hamilton graph • Explain the concept of cuts and cut vertices • Describe the Network flow problem • Working on Maxflow and Mincut theorem • Represent the graph and tress in computer • Find the application of Tress and graphs	4.7. Planarity Testing Algorithms 4.8. Eulerian Graph 4.9. Hamiltonian Graph 4.10. Graph coloring 4.11. Tree: Introduction and application, tree traversal, spanning tree, minimum spanning tree (Kruskal's algorithm) 4.12. Cutsets and Cutvertices 4.13. Network Flows, Maxflow and Mincut Theorem 4.14. Network Application of Trees and Graphs

4. **Methodology and Techniques**

Modes of instruction*:* Lecture, seminar, exercise course, guided personal study, tutorial, independent study, project work, Assignments in different topics, group discussion, reflective writing

Types of learning activities

Attending lectures, performing specific assignments, writing papers, independent and private study, reading books, journals and papers, providing constructive feedback, group study and peer discussion.

5. Evaluation Scheme

5.1 Internal Evaluation 40%

Internal Evaluation will be conducted by course teacher based on following activities.

a) **Attendance and Participation in class activities**: **5+5=10marks**

b) **Assignment I: Reflective Notes and Class presentation:** **5+5=10marks**

(Reflective notes on2to4 questions given by teacher at the end of the every unit and presentation on any two questions among them)

c) **Assignment II**: **One Term paper/Essay/Project and Interview:** **5+5=10marks**

(Logical essay /term paper /project on the topics chosen by students and approved by the teacher and interview)

d) **Mid-term exam**: **10marks**

5.2 External Evaluation (Final Examination) 40%

Types of questions	Total questions to be asked	Number of questions to be answered and marks allocated	Total marks
Group A: Multiple choice items	8 questions	8×1	8
Group B: Short answer questions	6 with 2 'or' questions	6×4	24
Group C: Long answer questions	1 with 1 'or' question	1×8	8

5.3 External Practical Evaluation (20%)

Office of the Controller of Examination will conduct final practical examination at the end of final examination.

After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the following evaluation criteria. There will be an internal examiner to assist the external examiner. Three hours' time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Evaluation System:

Practical	Weightage	Marks
Practical Report Copy	5	20
Viva	5	
Practical Exam	10	

Laboratory Works:

The laboratory work consists of implementing the algorithms and concepts discussed in the class. Student should implement problems with following concepts;

- Programs to implement Euclidean and Extended Euclidean algorithms.
- Programs to implement binary integer addition, multiplication, and division.
- Programs to generate truth tables of compound propositions.
- Programs to test validity of arguments by using truth tables.
- Programs to generate permutations and combinations.
- Programs to implements some probabilistic and randomized algorithms.
- Programs for representing relations, testing its properties, and testing equivalence.
- Programs to represent graphs, finding shortest path, and generating minimum spanning trees.
- Program for counting and some recursive algorithms.
- Algorithms for Relations, Graphs and tree.

Text Books:

Bernard, K., Robert B., & Sharon, C. R. (2015). *Discrete mathematical structures (6th ed.)*. Pearson Publications.

Joe, L. M., Kandel, A., & Baker, T. P. (2008). *Discrete mathematics for computer scientists and mathematicians (2nd ed.)*. Prentice Hall of India.

Kenneth, H. R. (2012). *Discrete mathematics and its applications (7th ed.)*. McGraw Hill Publication.

Reference Books:

Bogart, K., Drysdale, S., & Stein, C. (2010). *Discrete mathematics for computer scientists.* Addison-Wesley.

Far Western University
Faculty of Education
Bachelor in Computer Science Education

Course Title: Object Oriented Programming with Java Nature: Theory and Practical
Course No.: CS.Ed.234 Credits: 3
Level: B.Ed. Total Periods: 48+16
Third Semester Practical: 2 hours per period

---

1. Course Introduction

This course introduces the fundamental programming concepts and techniques in Java. All elements of object-oriented programming are introduced. Topics covered include control structures, classes and objects, dynamic memory allocation, Inheritance and Polymorphism, File Handling, Multithreading, Exception Handling.

2. Objectives

Upon completion of this course students should:

→ Understand the basic concepts and principles of object oriented programming.
→ Be able to design, write and test a Java program to implement a working solution to a given problem specification.
→ Be able to deal with exceptions effectively and write multithreaded programs

3. Specific Objectives and Contents

Specific Objectives	Contents
• Understand importance of java technology • Setup java environment and get ready for coding • Compile and Execute java programs • Know how to compile and run java program	**Unit I: Java Programming Basics (6 Hrs)** 1.1. History of java, Characteristics of java, Architecture of java 1.2. Java Virtual machine and byte code. 1.3. Procedure Oriented Vs Object Oriented Programming 1.4. Characteristic of object oriented Programming. 1.5. Setting up your computer for java environment (PATH and CLASSPATH Variables) , Structure of Java Programs, Compiling & Running Java Programs 1.6. Review of Data Types (Primitive and user defined), Comments, Operators (Arithmetic, relational, logical bitwise, assignment, conditional, shift, auto increment and auto decrement), variables, identifier and constant, Converting between Data Types (Type Casting), String (creation, concatenation, conversion of string, changing case, character extraction and string comparison), Array (creation and types)
• Use decision statements in programs • Demonstrate looping statements and	**Unit II: Control Flow (5 Hrs)** 2.1. Selection Statements: if statements, if....else statements,

program them • Apply jump statements in programs	else if ladders, switch statements 2.2. Looping: While Loop, Do While Loop, For Loop, Enhanced For Loop 2.3. Jump Statements: Break Statement, Continue Statement, Return Statement
• Understand class and objects and develop programs around it. • Use access Specifiers properly to class members • Exemplify static data members and methods • Implement constructors and use it in programs • Pass arguments and return values from methods	**Unit III: Class and Objects (7 Hrs)** 3.1. Creating Classes, Defining member variables and methods, Creating Reference Variables, Creating Objects, Using member variables and methods 3.2. Access Specifiers: Public, Protected, Default, and Private 3.3. Static and Non-static members, Constructors (default and parameterized), This Keyword, Garbage Collection, Inner Classes, Local Classes 3.4. Passing Parameters, Arrays, Objects to Methods and Constructors, Returning Values, Arrays, Objects from Methods and Constructors
• Write polymorphic programs using overloading and overriding • Understand importance of inheritance and use it in writing programs • Explain concepts of containership and abstract classes	**Unit IV: Inheritance and Polymorphism (7 Hrs)** 4.1. Method Overloading, Constructor Overloading, Creating Subclass, Different Types of Inheritance 4.2. Method Overriding, Dynamic Method Dispatch, Using Constructors and Inheritance, Super Keyword 4.3. Access Specifiers and Inheritance, Final Methods, Final Classes 4.4. Has-a Relationship (Containership), Object Class, Abstract Classes
• Understand interfaces and use it in programs • Differentiate between interfaces and abstract classes. • Demonstrate packages by creating and using it.	**Unit V: Interfaces and Packages (5 Hrs)** 5.1. Defining Interfaces, Interfaces vs. Classes, Extending Interfaces, Implementing Interfaces, Multiple Inheritance by using interfaces, Abstract Classes vs. Interfaces. 5.2. Package, Importance of Packages, Types of package (library and user defined), Using Packages, Creation of user defined packages
• Read inputs from files and store outputs in files. • Understand and use byte stream classes and character stream classes • Use random access and tokenizer in files	**Unit VI: File and I/O Handling (6 Hrs)** 6.1. Concept of IO Streams, File Class, Input Stream and Output Stream Class, File Input Stream and File Output Stream Class, Buffered Input Stream and Buffered Output Stream Class 6.2. Reader and Writer Classes, File Reader and File Writer Class, Input Stream Reader and Output Stream Writer Class. 6.3. Random File Access, Stream Tokenizer Class, Using Print Writer Class, Using Scanner Class
• Understand exceptions and its categories • Hand exceptional conditions in programs by using different keywords	**Unit VII: Exception Handling (6 Hrs)** 7.1. Concept of Exception and Exception Handling, Categories of Exceptions, Hierarchy of Exception Classes 7.2. Using Try....Catch, Multiple Catch Blocks, Finally Keyword

• Define own exception classes and use them in exception handling	7.3. Using Throws and Throw Keywords, Nested Try....Catch, Creating Exception Classes
• Explain importance of multithreaded programs • Use Runnable interface and Thread class in creating threads • Understand thread life cycle and manage multithreaded programs by using different methods.	**Unit VIII: Multithreading (6Hrs)** 8.1. Concept of Thread and Multithreading, Main Thread, Naming a Thread, Pausing a Thread, Thread Life Cycle 8.2. Multithreading by Using Runnable Interface, Multithreading by using Thread Class, Creating multiple threads, Joining Threads, setting Thread Priority, Stopping Threads 8.3. Thread Synchronization, Communication between Threads, Suspending and Resuming Threads

4. Methodology and Techniques

Modes of instruction*:* Lecture, seminar, exercise course, guided personal study, tutorial, independent study, project work, Assignments in different topics, group discussion, reflective writing

Types of learning activities
Attending lectures, performing specific assignments, writing papers, independent and private study, reading books, journals and papers, providing constructive feedback, group study and peer discussion.

5. Evaluation Scheme

5.2 Internal Evaluation40%
Internal Evaluation will be conducted by course teacher based on following activities.

e) **Attendance and Participation in class activities**: **5+5=10marks**

f) **Assignment I: Reflective Notes and Class presentation:** **5+5=10marks**
(Reflective notes on 2to4 questions given by teacher at the end of the every unit and presentation on any two questions among them)

g) **Assignment II**: **One Term paper/Essay/Project and Interview:** **5+5=10marks**
(Logical essay /term paper /project on the topics chosen by students and approved by the teacher and interview)

h) **Mid-term exam**: **10marks**

5.2 External Evaluation theory (Final Examination) 40%

Types of questions	Total questions to be asked	Number of questions to be answered and marks allocated	Total marks
Group A: Multiple choice items	8 questions	8×1	8
Group B: Short answer questions	6 with 2 'or' questions	6×4	24

Group C: Long answer questions	1 with 1 'or' question	1×8	8

5.3 External Practical Evaluation (20%)

Office of the Controller of Examination will conduct final practical examination at the end of final examination.

After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the following evaluation criteria. There will be an internal examiner to assist the external examiner. Three hours' time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Evaluation System:

Practical	Weightage	Marks
Practical Report Copy	5	20
Viva	5	
Practical Exam	10	

Laboratory Work

Student should write programs and prepare lab sheet for most of the units in the syllabus. They should practice design and implementation of java programs that demonstrates different concepts discussed is class. However, nature of programming can be decided by the instructor. The lab work should be practiced for minimum of 3 lab hours per week.

Text Books and Reference Materials

1. Horstmann, C.S., & Cornell, G. (2012). *Core Java Volume I-Fundamentals (9th ed)*. Prentice Hall (All units). https://www2.nsru.ac.th/tung/java_doc
2. Deitel, P., & Deitel, H. (2011). *Java: How to program (9th ed.).* https://docs.google.com/file/
3. Hebert, S. J. (2014.). *The Complete Reference* (9th ed). McGraw-Hill Education.
4. Steven, H. J. (2013). *Java 7 Programming Black Book.* Dreamtech Press.

Far Western University
Faculty of Education
Bachelor in Computer Science Education

Course Title: Operating System
Course No.: CS.Ed.235
Third Semester
Level: B.Ed.

Nature: Theory and Practical
Theory Period Per Week: 3
Total Periods: 48+16
Practical: 2 hours per period

1. Course Description

The Operating Systems course aims to provide students with a comprehensive understanding of the principles, design, and management of operating systems. Students will gain Theoretical and practical knowledge and skills necessary for developing, managing, and troubleshooting operating systems in various computing environments.

2. Specific Objectives and Contents

Specific Objectives	Contents
• Define the term operating system • Identify the key components of operating system • List and describe the primary functions of operating system. • Trace the historical development of operating system. • Classify the operating system on different basis • Explain the different types of system calls • Define user space and kernel space in the context of OS • Compare and contrast the architectural structures of OS	**Unit I: Overview of Operating System (10)** 1.1. Introduction to Operating System 1.2. Functions of Operating System 1.3. History of Operating System. 1.4. Types of Operating System 1.5. Operations of Operating System 1.6. Introduction to System Calls: Calls for process management, Calls for File Management, Calls for protection. 1.7. Introduction to User space and Kernel space 1.8. Structure of Operating System: Monolithic, Layered, Virtual Machines and Client server model **Lab Work:** Install and Configure Different Operating Systems (Windows, Linux, MINIX). Explore the Basic Commands of Command-Line Interfaces (CLI) in Different Operating Systems.
• Define the concept of process • Describe the process model and its components • Explain the relationships between process, program and thread • Discuss the process operations and states	**Unit II: Process Management (11)** 2.1. Process Concept 2.2. Process Model 2.3. Process Operations and sates 2.4. Implementation of Process: Process Control Block

• Define inter-process communication (IPC). • Identify the need for IPC in a multitasking environment. • Describe different types of IPC mechanisms • Define race conditions • Explain the potential issues in race condition • Discuss the techniques to avoid the race conditions • Outline the criteria used for process scheduling • Compare and contrast between different types of scheduling algorithms • Define deadlock • Identify the different conditions for deadlock • Describe the Banka's algorithms for deadlock detection and avoidance	2.5. Threads: Process vs Threads, Types of Threads 2.6. Inter-process Communication: Introduction, Types of IPC 2.7. Race Conditions and Ways to Avoid Race Conditions 2.8. Process Scheduling: Scheduling Criteria, Scheduling Algorithms: FCFC(First Come First Serve), Shortest Job First(SJF), Round Robin Algorithm and Priority Scheduling. 2.9 Introduction to Multiple Processor Scheduling 2.10. Deadlock and Avoidance: Introduction to Deadlock, Necessary Condition for Deadlock, Resource Allocation Graph, Deadlock Detection (Banker's Algorithm), Recovery and Prevention. **Lab Work:** Write Programs to Create and Manage Processes in single and Multi-Process Environment. Implement Various Process Scheduling Algorithms and Analyze their Performance (Project Work).
• Define the concept of memory hierarchy • Explain the concept of memory management and its significance • Define mono-programming and explain its limitations. • Describe memory management in a mono-programming environment without swapping and paging. • Explain the memory management with swapping • Define virtual memory and its role in memory management • Explain the concept of virtual memory management techniques • Explain the concept of paging and how it is implemented. • Compare and contrast different page replacement algorithms such as FIFO, Second Chance, and LRU. • Describe memory management with segmentation and its advantages over virtual memory management	**Unit III: Memory Management (9)** 3.1. Introduction to Memory Hierarchy 3.2. Basic Memory Management Techniques 3.2.1. Mono-programming Without Swapping and Paging 3.2.2. Multiprogramming with Fixed Partition 3.3. Swapping: Memory Management with bit map and linked list 3.4 Virtual Memory Management: Paging, Page table and Page Replacements Algorithms (FIFO, second chance and LRU) 3.5 Memory Management with Segmentation **Lab Work**: Simulate the concept of Virtual Memory Management Implement page replacement algorithms and analyses their performance (Project Work). .
• Define the concept of file • Explain the different file operations • Describe the importance of file system in computer system • Explain different file organization and access methods • Compare and contrast different directory	**Unit IV. File System (5)** 4.1 File Concept: Introduction, operations on File and File attributes 4.2 File Organization: Sequential, Random, Indexed, Hashed Organization 4.3 Access Methods

structures • Compare and contrast FAT and NTFS file system	4.3 Directory Structure: Single level, Two level and Tree structure 4.3 File System Implementation: File Allocation Methods, FAT and NTFS file system **Lab Work:** Implement Basic File System Operations: Create, Read, Write, and Delete. Simulate the concept of file organization
• Define input output subsystem • Explain different mode of I/O in computer system • Explain the role of device drivers • Describe the structure of disk and its components • Compare and contrast the disk scheduling algorithms	**Unit V: Input Output System(5)** 5.1 I/O System Architecture: Isolated vs memory mapped I/O, direct memory access(DMA) 5.2 I/O Devices and Drivers 5.3 Disk Structure and Disk Scheduling: Scheduling Algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK **Lab Work:** Implement the various disk scheduling algorithms and analyses their performance (project).
• Define the terms protection and security • Explain the different security threats • Discuss the guidelines for designing the security principles • Define the terms authentication and authorization • Explain the different authentication and authorization methods • Define the concept of cryptography and explain its roles in maintaining the security • Identify the different security measures in operating system	**Unit VI: Protection and Security(5)** 6.1 Security Threats and Policies: Malware, Network threat, Physical security threat, policy guidelines. 6.2 Authentication and Authorization 6.2.1 Authentication Methods: Passwords, biometrics, two-factor authentication. 6.2.2 Access Control Models 6.3 Cryptography in Operating Systems: Introduction, symmetric and asymmetric key cryptography. 6.4 Security Measures in OS: Firewalls, Antivirus, and security update. **Lab Work:** Configure and Implement Basic Security Measures on Operating Systems. Develop Programs to Implement Authentication and Authorization Mechanisms.
• Identify and explain the key features and functionalities of Windows and Linux operating systems. • Demonstrate proficiency in executing basic Linux commands for file manipulation, navigation, and system administration. • Compare and contrast the command-line interfaces of Windows and Linux. • Explain the emerging concept of operating system • Critically evaluate the effectiveness and potential limitations of implementing	**Unit VII: Case study (3)** 7.1 Case study on Window's, LINUX: Working on LINUS commands 7.2 Emerging concept in operating system: Server less computing, Edge computing, green computing. 7.3 Cloud Operating System 7.4 Mobile Operating System **Lab Work:** Analyze the popular operating system and prepare a short report on their importance, similarities, differences,

these emerging concepts in operating systems, drawing insights from case studies. • Understand the role of cloud operating systems in cloud computing environments and their impact on resource management and scalability. • Identify and describe the characteristics of mobile operating systems and their role in supporting mobile device functionality.	applications and security principles. Work on basic DOS and Linux shell scripting

3. Methodology and Techniques

Modes of instruction: Lecture, seminar, exercise course, guided personal study, tutorial, independent study, project work, Assignments in different topics, group discussion, reflective writing

Types of learning activities

Attending lectures, performing specific assignments, writing papers, independent and private study, reading books, journals and papers, providing constructive feedback, group study and peer discussion.

4. Evaluation Scheme

4.1 Internal Evaluation40%

Internal Evaluation will be conducted by course teacher based on following activities.

i) **Attendance and Participation in class activities**: **5+5=10marks**

j) **Assignment I: Reflective Notes and Class presentation:** **5+5=10marks**
(Reflective notes on 2 to 4 questions given by teacher at the end of the every unit and presentation on any two questions among them)

k) **Assignment II**: **One Term paper/Essay/Project and Interview:** **5+5=10marks**
(Logical essay /term paper /project on the topics chosen by students and approved by the teacher and interview)

l) **Mid-term exam**: **10marks**

4.2 External Evaluation (Final Examination) 40%

Types of questions	Total questions to be asked	Number of questions to be answered and marks allocated	Total marks
Group A: Multiple choice items	8 questions	8×1	8
Group B: Short answer questions	6 with 2 'or' questions	6×4	24
Group C: Long answer questions	1 with 1 'or' question	1×8	8

4.3 External Practical Evaluation (20%)

Office of the Controller of Examination will conduct final practical examination at the end of final examination.

After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the following evaluation criteria. There will be an internal examiner to assist the external examiner. Three hours' time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Evaluation System:

Practical	Weightage	Marks
Practical Report Copy	5	20
Viva	5	
Practical Exam	10	

Prescribed Books

Textbooks

Andrew, S. T., & Herbert, B. (2015). *Modern operating systems (4th ed.)*. Pearson Education.

References

Silberschatz, A., & Galvin, P. B., & Gagne, G. (2013). *Operating System Concepts (9th ed.)*. Wiley.