

II/IV B. TECH. SECOND SEMESTER**FILE STRUCTURES****(Required)****Course Code : CS 4T3****Credits: 3****Lecture: 3 periods/ week****Internal assessment: 30 Marks****Tutorial: 1period/week****Semester end examination: 70 Marks****Prerequisites: Data Structures**

Course Objectives:

1. Provide an introduction to the fundamental file operations and storage systems.
2. Introducing fundamental concepts of file structure.
3. Introducing the most important high-level file structures tools which include indexing, co sequential processing, B trees, Hashing.
4. Applying the techniques in the design of C++ programs for solving various file management problems.

Course Outcomes:

At the end of this course student will:

CO1) Understand the fundamental concepts of file processing operations and storage structures

CO2) Apply object orientation concepts to manipulate records

CO3) Apply concepts of sorting and merging on multiple files

CO4) Analyze the sequential and indexing file accessing techniques with appropriate data structures

CO5) Illustrate the usage of hashing techniques to organize file structures

Syllabus:**UNIT 1**

Fundamental File Processing Operations : Physical Files and Logical Files, Opening Files, Closing Files, Reading and Writing, Seeking, Special Characters in Files, The Unix Directory Structure, Physical Devices and Logical Files.

Secondary Storage and System Software: Disks, Magnetic Tape, Disk versus Tape,

Introduction to CD-ROM, Physical Organization of CD-ROM, CD-ROM Strengths and Weaknesses; Storage as a Hierarchy, A Journey of a Byte, Buffer Management, I/O in UNIX.

UNIT 2

Fundamental File structure Concepts: Field and Record Organization, Using Classes to Manipulate Buffers, Using Inheritance for Record Buffer Classes, Managing Fixed Length and Fixed Field Buffers.

Managing Files of Records: Record Access, More about Record Structures, Encapsulating Record I/O Operations in a Single Class, File Access and File Organization.

Co-sequential Processing: An Object-Oriented Model for Implementing Cosequential Processes.

UNIT 3

Indexing: What is an Index?, A Simple Index for Entry -Sequenced Files. **Multilevel**

Indexing and B-Trees: Introduction: The Invention of B-Tree, Statement of the Problem, Indexing with Binary Search Trees, Multilevel Indexing, a Better Approach to Tree Indexes, B-trees: Working up from the Bottom, Example of Creating a B-Tree, B-Tree Methods Search, Insert and Others.

UNIT 4

Indexed Sequential File Access and Prefix B+ Trees: Indexed Sequential Access, Maintaining a Sequence Set, Adding a Simple Index to the Sequence Set, The Content of the Index: Separators Instead of Keys, The Simple Prefix B+ Tree, Simple Prefix B+ Tree Maintenance.

UNIT 5

Hashing: Introduction, A Simple Hashing Algorithm, Hashing Functions and Record Distributions, How Much Extra Memory Should Be Used?, Collision Resolution by Progressive Overflow, Storing More Than One Record per Address: Buckets, Making Deletions, Other Collision Resolution Techniques, Patterns of Record Access.

Learning Resource

Text Books

1. File Structures: An Object-Oriented Approach with C++, Michael J. Folk, Greg Riccardi, Bill Zoellick, Third Edition, Pearson Education.

References

1. Data Management and File Structures, Mary E.S. Loomis, Second Edition, PHI.
2. File Organization and Processing, Alan L. Tharp, Wiley India Edition.