

BCA I Year II Semester

OPERATING SYSTEMS

Course Objectives

1. To understand the services provided by and the design of an operating system.
2. To understand the structure and organization of the file system.
3. To understand what a process is and how processes are synchronized and scheduled.
4. To understand different approaches to memory management.
5. Students should be able to use system calls for managing processes, memory and the file system.
6. Students should understand the data structures and algorithms used to implement an OS.

Course Outcomes

1. Analyze the concepts of processes in operating system and illustration of the scheduling of processor for a given problem instance.
2. Identify the dead lock situation and provide appropriate solution so that protection and security of the operating system is also maintained.
3. Analyze memory management techniques, concepts of virtual memory and disk scheduling.
4. Understand the implementation of file systems and directories along with the interfacing of IO devices with the operating system.

UNIT - I

Operating System Introduction: Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Parallel, Distributed Systems, Real-Time Systems, Operating System services.

UNIT - II

Process and CPU Scheduling - Process concepts - The Process, Process State, Process Control Block, Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Case studies: Linux, Windows.

Process Coordination - Process Synchronization, The Critical section Problem, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT - III

Memory Management and Virtual Memory - Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table. Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames.

UNIT - IV

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Structure,

Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling.

UNIT - V

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

REFERENCES BOOKS:

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition.
2. Principles of Operating Systems by Naresh Chauhan, OXFORD University Press
3. Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.
4. Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.
5. Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhare, TMH.
6. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
7. Operating Systems, A. S. Godbole, 2nd Edition, TMH

Student Activity:

- 1. Load any new operating system into your computer.**
 - 2. Partition the memory in your system**
 - 3. Create a semaphore for process synchronization**
-

Operating Systems Lab

1. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)
 2. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time. (2 sessions)
 3. Developing applications using Inter Process Communication (using shared memory, pipes or message queues)
 4. Implement the Producer – Consumer problem using semaphores
 5. Implement any two memory management schemes
 6. Implement any two file allocation techniques (Linked, Indexed or Contiguous)
 7. Implement any two Page Replacement Algorithms
 8. Implement Deadlock prevention algorithm.
 9. Implement any two disk scanning algorithms
-

BCA I Year II Semester

OBJECT ORIENTED PROGRAMMING USING C++

Course Objectives

This course covers object-oriented programming principles and techniques using C++. Topics include pointers, classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and low-level language features. This course also covers basic concepts for software design and reuse.

Course Outcomes

1. Understand concepts of objects and their significance in real world
2. Investigate software problem in terms of objects and entities
3. Learn to co-relate relationship among different entities involved in a system
4. Find dependency and roles in an environment
5. Develop software in terms of objects, associations, and integrity constraints
6. Generalize and aggregate business entities and transform behavior into functions
7. Identify, understand and analyze various sample development models

UNIT I

Principles of OOP: Software Crisis. Software Evolution- Programming Paradigms. Object Oriented Technology- Basic concepts and benefits of OOP – Application of OOP, OOP languages

Introduction to C++: History of C++, Structure of C++, Application of C++, tokens, keywords, identifiers, basic data types, derived data types, derived data types, symbolic constant, dynamic initialization, reference variables, scope resolution operator, type modifiers, type casting operators and control statements, input and output statements in C++, Function prototyping and components, Passing parameters: Call by reference, Return by reference, Inline function, Default arguments, Over loaded function.

UNIT II

Classes and Objects: Class specification, Member function definition – nested member function, access qualifiers, static data members and, member functions. Instance creation - Array of objects - Dynamic objects - Static Objects – Objects as arguments -Returning objects

Constructors and Destructors: Constructors- Parameterized constructors, Overloaded Constructors, Constructors with default arguments, copy constructors, Destructors.

UNIT III

Operator Overloading: Operator function-overloading unary and binary operators, overloading the operator using Friend function, Stream operator overloading, Data conversion.

Inheritance: Defining derived classes. Single Inheritance - Protected data with private inheritance - Multiple Inheritances - Multi Level Inheritance - Hierarchical Inheritance. Hybrid

Inheritance - Multi path Inheritance - Constructors in derived and base Class -Template in Inheritance - Abstract classes - Virtual function and Dynamic polymorphism. -
Virtual destructor - Nested Classes

UNIT- IV

Functions in C++ : Virtual functions- need for Virtual function, , Pure Virtual functions, Generic Programming with Templates. Introduction, function templates, overloaded function templates, user defined templates arguments, class templates, Inheritance of class templates.

UNIT-V

Files: file stream, file pointer and manipulation, file open and close, sequential and random access.

Exception Handling: Principle of Exception handling, Exception handling mechanism ,Multiple catch, Nested try, re throwing the Exception.

REFERENCE BOOKS:

- 1.1 Object Oriented Programming with C++ by Reema Thareja, OXFORD University Press
- 2.The Complete Reference C++, Herb Schildt, Tata McGraw-Hill, Fourth Edition.
- 3.Robert Lafore, "Object Oriented Programming in C++", Galgotia Publication Pvt. Ltd,4 th edition, New Delhi, 2002
- 4.Ashok N Kamathane, "Object Oriented Programming with ANSI & Turbo C++", Pearson Education, New Delhi, 2003.
- 5.Bjarne Stroustrup," C++ Programming language", Pearson Education, New Delhi, 2001.
- 6.Venugopal K R, Rajkumar Buyya and Ravishankar T," Mastering C++", TMH, ND, 2006

Student Activity:

1. Create a class diagram for academic process in your college
 2. Write a program to implement "Vikuntapali"game
-

OBJECT ORIENTED PROGRAMMING USING C++ LAB

1. Write a C++ program to find the sum of individual digits of a positive integer.
 2. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.
 3. Write a C++ program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
 4. Write a C++ program to find the factorial of a given integer
 5. Write a C++ program to find the GCD of two given integers
 6. Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.
 7. Write a C++ program to implement call by value and call by reference parameters passing
 8. Write a C++ program to implement function templates
 9. Write a program to implement Overloading and Overriding
 10. Write a C++ program to implement the matrix ADT using a class. The operations supported by this ADT are:
 - a. Reading a matrix.
 - b. Printing a matrix
 - c. Addition of matrices
 - d. Subtraction of matrices
 - e. Multiplication of matrices
 11. Write C++ programs that illustrate how the Single inheritance, Multiple inheritance Multi level inheritance and Hierarchical inheritance forms of inheritance are supported
 12. Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class
 13. Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions
-

Adobe In Design LAB

Implement the following tasks using ADOBE

- 1. Resume designing**
 - 2. Paragraph setting**
 - 3. Text column wise designing**
 - 4. Text base paper add**
 - 5. Create college Logo**
 - 6. Table creation**
 - 7. Student marks list**
 - 8. Book work**
 - 9. Picture insertion**
 - 10. Application form**
 - 11. Text based Visiting card**
 - 12. Notice designing**
 - 13. Typographic alignment styles**
 - 14. Wedding card designing**
 - 15. Letter models**
-