BCA 3rd semester Syllabus w.e.f. 2025-26 onwards

Paper 1	**Operating System Concepts**	Credits: 4	Contact Hours: 52	Theory 04 Hrs/week
Internal assessment: 20 marks		Term end exam: 80 marks	Exam duration: 03 hrs	

Course Outcomes (COs): At the end of the course, students will be able to:

- Explain the structure and functions of the operating system.
- Comprehend multithreaded programming, process management, process synchronization, memory management and storage management.
- Compare the performance of Scheduling Algorithms
- Identify the features of I/O and File handling methods

UNIT 1 **10 Hrs**

Introduction to Operating System: Definition, History and Examples of Operating System; Computer System organization; Types of Operating Systems; Functions of Operating System; Systems Calls; Operating System Structure.

UNIT 2 **10 Hrs**

Process Management: Process Concept- Process Definition, Process State, Process Control Block, Threads; Process scheduling- Multiprogramming, Scheduling Queues, CPU Scheduling, Context Switch; Operations on Processes Creation and Termination of Processes; Inter process communication (IPC)- Definition and Need for Inter process Communication; IPC Implementation Methods- Shared Memory and Message Passing;

UNIT 3 **10 Hrs**

Multithreaded Programming: Introduction to Threads; Types of Threads; Multithreading- Definition, Advantages; Multithreading Models; Thread Libraries; Threading Issues. CPU Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling; Multiprocessor Scheduling; Real-Time CPU Scheduling.

UNIT 4 **12 Hrs**

Process Synchronization: Introduction; Race Condition; Critical Section Problem and Peterson's Solution; Synchronization Hardware, Semaphores; Classic Problems of Synchronization- Readers and Writers Problem, Dining Philosophers Problem; Monitors. Deadlocks: System Model; Deadlocks Characterization; Methods for Handling Deadlocks; Deadlock Prevention; Deadlock Avoidance; Deadlock Detection; and Recovery from Deadlock.

UNIT 5 **10 Hrs**

Memory Management: Logical and Physical Address Space; Swapping; Contiguous Allocation; Paging; Segmentation; Segmentation with Paging. Virtual Memory: Introduction to Virtual Memory; Demand Paging; Page Replacement; Page Replacement Algorithms; Allocation of frames, Thrashing

Text Book

1. Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne., 9th Edition, Wiley, 2012.

References:

2. Operating System Concepts - Engineering Handbook, Ghosh PK, 2019.
3. Understanding Operating Systems, McHoes A et al., 7 th Edition, Cengage Learning, 2014.
4. Operating Systems - Internals and Design Principles, William Stallings, 9th Edition, Pearson Education.
5. Operating Systems – A Concept Based Approach, Dhamdhere, 3rd Edition, McGraw Hill Education India.
6. Modern Operating Systems, Andrew S Tanenbaum, 4^{th} Edition, Pearson Education.

<table>
<tr><td colspan="2">Paper-1Lab</td><td colspan="2">LAB: Operating System Lab</td><td>Credits: 2</td><td>Contact Hours: 52</td><td>Practical 04 Hrs/week</td></tr>
<tr><td colspan="3">Internal assessment: 10 marks</td><td colspan="2">Term end exam: 40 marks</td><td colspan="2">Exam duration: 02 hrs</td></tr>
</table>

Assignments based on the subject Paper-1: **Operating System Concepts** shall be implemented in the lab.

1. Basic UNIX Commands and various UNIX editors such as vi, ed, ex and EMACS
2. UNIX and Windows File manipulation commands
3. C Program For System Calls Of Unix Operating Systems (Opendir, Readdir, fork, getpid, exit)
4. C programs to simulate UNIX commands like cp, ls, grep.
5. Simple shell programs by using conditional, branching and looping statements (to check the given number is even or odd,the e given year is leap year or not, find the factorial of a number, swap the two integers)
6. To write a C program for implementation of Priority scheduling algorithms
7. To write a C program for implementation of Round Robin scheduling algorithms.
8. To write a C program for implementation of SJF scheduling algorithms
9. To write a C-program to implement the producer – consumer problem using semaphores
10. To write a C program to implement banker‟s algorithm for deadlock avoidance.
11. To write a c program to implement Threading and Synchronization Applications.
12. To write a C program for implementation of memory allocation FCFS and SJF scheduling algorithms.
13. To write a c program to implement Paging technique for memory management.
14. To write a c program to implement semaphores.
15. To write a c program to implement Bankers algorithm.

Evaluation Scheme for Lab. Term end Examination

Assessment Criteria		Marks
Program– 1	Writing the Program	05
	Execution and Formatting	05
Program– 2	Writing the Program	05
	Execution and Formatting	05
Viva Voice		05
Practical Record book		05
	Total	**40**

Paper 2	Design and Analysis of Algorithms	Credits: 4	Contact Hours: 52	Theory 04 Hrs/week
Internal assessment: 20 marks		Term end exam: 80 marks	Exam duration: 03 hrs	

Course Outcomes (COs): At the end of the course, students will be able to:

- the time complexity of an algorithm and asymptotic notation is used to provide classification of algorithms
- understand different computational models and their complexity(e.g., divide-and-conquer, greedy algorithms, dynamic programming, etc)
- analyze and design algorithms and the impact of algorithm design in practice
- write program to execute and analyze different algorithms
- understand the concepts of deterministic and non-deterministic algorithms.

UNIT 1 **10 Hrs**

Design of Efficient Algorithms and Elementary Data Structures: Algorithm specification, Performance analysis, Time and Space Complexity, Asymptotic notation, Review of Stack, Queues, Trees. Operations on Stack, Queue and Trees. Recursion, Heaps and Heap Sort

UNIT 2 **12 Hrs**

Divide and Conquer: General Method, Binary Search, Max and Min, Merge Sort, Quick Sort, Matrix Multiplication and Related Operations; Strassen's Matrix Multiplication, Boolean Matrix Multiplication.

UNIT 3 **10 Hrs**

The Greedy Method: The General Method, Knapsack Problem, Job Sequencing with Deadlines, Minimum Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm. Single Source Shortest Paths

UNIT 4 **10 Hrs**

Dynamic Programming: The General Method, Multistage Graphs, All Pair's Shortest Paths, 0/1 knapsack, Travelling Salesman Problem

UNIT 4 **10 Hrs**

Backtracking: General Methods, 8 – Queens Problem, Sum of Subsets, Knapsack Problem, NP – Hard and NP – Complete Problems.

Text Book:

1. Ellis, Horwitz, SartajSahani and Rajashekaran S., “Computer Algorithms”, (1999) Galgotia Publications Pvt.,Ltd.

Reference Books:

1. Aho A.V, Hopcroft J.E and Ullman, J.D., “The Design and Analysis of Computer Algorithms”, (1976) Addison – Wesley.
2. Introduction to Algorithms, (2009) third edition, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, MIT press.
3. Sara Baase, Computer Algorithms, “An Introduction to Design and Analysis”, Addison Wesley.

4. Allen Weiss (2009), Data structures and Algorithm Analysis in C++, 2nd edition, Pearson education, New Delhi.
5. R. C. T. Lee, S. S. Tseng, R.C. Chang and T. Tsai (2006), Introduction to Design and Analysis of Algorithms A strategic approach, McGraw Hill, India.

Paper-2 Lab	**LAB: Algorithms Lab. Using C/JAVA**	Credits: 2	Contact Hours: 52	Practical 04 Hrs/week
Internal assessment: 10 marks		Term end exam: 40 marks		Exam duration: 02 hrs

Assignments based on the subject Paper-2: **Introduction to Algorithms Design** shall be implemented in the lab.

1. Program to construct a stack of elements and to perform the following operations on: push, pop, status, empty, full, display
2. Program to construct a queue of integers and to perform following operations on it: enqueuer, de-queue, status, empty, full
3. Program to implement
   a. Binary Search,
   b. Max and Min,
   c. Merge Sort,
   d. Quick Sort,
   e. Strassen's Matrix Multiplication,
   f. Boolean Matrix Multiplication
4. Program to implement
   a. Job Sequencing with Deadlines,
   b. Prim's Algorithm,
   c. Kruskal's Algorithm
5. Program to implement All Pair's Shortest Paths
6. Dijkstras Algorithm
7. Program to implement all pair shortest problem using dynamic programming technique
8. Program to solve Travelling Salesman Problem

Evaluation Scheme for Lab. Term end Examination

Assessment Criteria		Marks
Program– 1	Writing the Program	05
	Execution and Formatting	05
Program– 2	Writing the Program	05
	Execution and Formatting	05
Viva Voice		05
Practical Record book		05
	Total	**40**

Paper 3	**Theory of Computation**	Credits: 4	Contact Hours: 52	Theory 04 Hrs/week
Internal assessment: 20 marks		Term end exam: 80 marks	Exam duration: 03 hrs	

Course Outcomes (COs): At the end of the course, students will be able to:

- understand basic concepts of formal languages of finite automata techniques
- Design Finite Automata's for different Regular Expressions and Languages
- Construct context free grammar for various languages
- solve various problems of applying normal form techniques, push down automata and Turing Machines .

Unit 1 **10 Hrs**

Introduction to formal proof – Additional forms of proof – Inductive proofs –Finite Automata (FA) – Deterministic Finite Automata (DFA) – Non-deterministic Finite Automata (NFA) – Finite Automata with Epsilon transitions.

Unit 2 **10 Hrs**

Regular Expression – FA and Regular Expressions – Proving languages not to be regular – Closure properties of regular languages – Equivalence and minimization of Automata.

Unit 3` **12 Hrs**

Context-Free Grammar (CFG) – Parse Trees – Ambiguity in grammars and languages – Definition of the Pushdown automata – Languages of a Pushdown Automata – Equivalence of Pushdown automata and CFG– Deterministic Pushdown Automata.

Unit 4 **10 Hrs**

Normal forms for CFG – Pumping Lemma for CFL – Closure Properties of CFL – Turing Machines – Programming Techniques for TM.

Unit 5 **10 Hrs**

A language that is not Recursively Enumerable (RE) – An undecidable problem that is RE – Undecidable problems about Turing Machine – Post's Correspondence Problem – The classes P and NP.

Text Book

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", second Edition, Pearson Education, 2007.

References:

1. H.R. Lewis and C.H. Papadimitriou, "Elements of the theory of Computation", Second Edition, Pearson Education, 2003.
2. Thomas A. Sudkamp," An Introduction to the Theory of Computer Science, Languages and Machines", Third Edition, Pearson Education, 2007.
3. Raymond Greenlaw an H.James Hoover, " Fundamentals of Theory of Computation, Principles and Practice", Morgan Kaufmann Publishers, 1998.
4. Micheal Sipser, "Introduction of the Theory and Computation", Thomson Brokecole, 1997

BCA 3th semester Syllabus w.e.f. 2026-27 onwards

Paper-1Lab	LAB: Theory of Computation	Credits:2	ContactHours:52	Practical04Hrs/week
Internalassessment:10 marks		Termendexam:40marks		Examduration:02hrs

Course Outcomes(COs):

Programs for practical

PART-A

1. 1. Design a DFA that accepts all strings over {0,1} ending with '01'.
2. 2. Construct an NFA for the language of strings over {a,b} containing substring 'ab'.
3. 3. Convert a given NFA into an equivalent DFA.
4. 4. Minimize a given DFA using the state minimization algorithm.
5. 5. Construct a DFA for the language of binary strings divisible by 3.
6. 6. Construct a regular expression for the language of strings over {a,b} with even number of a's.
7. 7. Convert a given regular expression into an NFA using Thompson's construction.
8. 8. Prove that the language $L = \{a^n b^n \mid n \geq 0\}$ is not regular using pumping lemma.
9. 9. Design a context-free grammar (CFG) for palindromes over {0,1}.
10. 10. Convert a given CFG into Chomsky Normal Form (CNF).

PART-B

11. Construct a PDA for the language $L = \{a^n b^n \mid n \geq 0\}$.
12. Construct a PDA for the language $L = \{ww^R \mid w \in \{a,b\}^*\}$.
13. Prove that the language $\{a^n b^n c^n \mid n \geq 0\}$ is not context-free.
14. Design a Turing Machine to accept the language $L = \{a^n b^n \mid n \geq 0\}$.
15. Construct a Turing Machine to compute the function f(n) = n+1 in unary representation.
16. Construct a Turing Machine to decide whether a binary string has equal number of 0's and 1's.
17. Show that the Halting problem is undecidable.
18. Demonstrate closure properties of regular languages with suitable examples.
19. Demonstrate closure properties of context-free languages with suitable examples.
20. Construct a Turing Machine to recognize palindromes over {0,1}
    1. and perform operations on DataFrames.

Evaluation Scheme for Lab. Term end Examination

Assessment Criteria		Marks
Program–1	Writing the Program	05
	Execution and Formatting	05
Program–2	Writing the Program	05
	Execution and Formatting	05
Viva Voice		05
Practical Record book		05
	Total	**40**