BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-I)
5th SEMESTER

CAP522J1: COMPUTER APPLICATIONS _ OPERATING SYSTEM
CREDITS: THEORY (3) PRACTICAL (1)

COURSE LEARNING OUTCOMES:
At the end of this course, a student should be able to:

- *To learn and understand the concepts of operating system.*
- *To understand the core structure and functions of operating system.*
- *Explain process management, processor scheduling, and concurrent programming.*
- *Understand the concept of deadlocks and synchronization.*
- *Learn memory management, file management. I/O systems, and disk scheduling.*
- *Distinguish main memory and virtual memory.*
- *Leam Operating System Design issues.*
- *Leam basic Unix commands and shell programming*

UNIT-1 (15 HOURS)
Introduction: Goals & Structure of Operating System. Basic functions & Modes. Types of Operating System (Batch, Multiprogramming, Multitasking, Time Sharing, Parallel, Distributed A Real-Time OS). System calls.

Process: Process Concept, Process states, Threads, Uni-processor Scheduling: Types of scheduling (Pre-emptive & Non- pre-emptive). Scheduling algorithms (FCFS, SJF, RR, Priority, Thread Scheduling).

UNIT-2 (15 HOURS)
Concurrency: Principles of Concurrency, Mutual Exclusion: Software approaches, Hardware Support, Semaphores, Message Passing, Signals. Monitors, Classical Problems of Synchronization (Readers-Writers, Producer-Consumer, and Dining Philosopher problem).

Deadlock: Principles of Deadlock. Deadlock Prevention & Avoidance, Deadlock Detection.

UNIT-3 (15 HOURS)
Memory: Memory Management requirements, Memory partitioning (Fixed and Variable Partitioning), Memory Allocation Strategies (First Fit, Best Fit and Worst Fit), Fragmentation. Swapping, Segmentation, Paging.

Virtual Memory: Virtual Memory Concepts and Implementation, Concept of Demand Paging, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies). Thrashing.

PRACTICALS (I CREDIT: 30 HOURS)

- In Malling and setting up of Linux /Unix Operating system.
- Using Basic Unix commands like echo, printf. Is, who, date, passwd, cal.
- Using pwd, cd, mkdir, rmdir, cat commands
- Using mv, rm, cp, wc and od commands.
- Demonstrate the use of Pipes.
- Using Basic and Extended regular expressions
- Implementing file permissions, printing commands.
- Mount devices like usb drives in UNIX/Linux.
- Write a shell script to demonstrate the use of variable & basic input output on the console.
- Write a script to demonstrate the use of control statements

TEST BOOK:

- *Peter B Galvin, Greg Gagne. Abraham Silberschatz: Operating System Concepts. John Wiley & Sons, Inc.*
- *Shell Programming by Yashwant Kanetkar*

REFERENCE BOOKS:

- *Peter B Galvin. Greg Gagne. Abraham Silberschatz: Operating System Concepts. John Wiley A Sons, Inc.*
- *Andrew S. Tanenbaum: Modern Operating Systems, PHI*
- *William Stallings: Operating Systems, Pearson Education India*
- *M. G. Venkatesh Murthy: USIX & Shell Programming, Pearson Education*
- *Richard Blum, Christine Bresnahan: Linux Command Line and Shell Scripting Bible, Wiley.*

BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-2)
5TH SEMESTER

CAP522J2: COMPUTER APPLICATIONS _ DATA STRUCTURES USING C
CREDITS: THEORY (4) INTERNSHIP / PRACTICAL (2)

OBJECTIVES:

- *To introduce the fundamentals of Data Structures, Abstract concepts and how these concepts are useful in problem solving.*
- *To learn the linear and non-linear data structures.*
- *To explore the applications of linear and non-linear data structures.*
- *To learn to represent data using tree and graph data structure.*
- *To learn the basic sorting and searching algorithms.*
- *To write programs for different Data Structures and Algorithms*

Unit-I: Linear Data Types – I (15 Lectures)
Introduction to data structure (Linear, Non-Linear, Primitive, Non-Primitive), Data Structure and Data Operations, Algorithm Complexity
Single dimensional array and its operations (Searching, traversing, inserting, deleting), Two-Dimensional array, Addressing, Sparse Matrices, recursion.
Searching Algorithms: Linear Search, Binary Search and their Comparison.

Unit –II: Linear Data Types – II (15 Lectures)
Array Sorting Algorithms: Selection sort. Insertion sort, Bubble Sort, Quick sort.
Stack: Definition & Concepts, Array Representation of Stack, Operations on Stack. Applications of Stack: Expressions and their representation: Infix, Prefix, Postfix and their conversions & evaluation.
Queue: Definition & Concepts, Array Representation of Queue, Operations on Queue, Circular Queue, Applications of Queue.

Unit-III: Linear Data Types – III (15 Lectures)
Review of structures & pointers.
Introduction to Linked Lists and their applications: Singly linked list: Definition & Concepts, representation in memory, operations on singly linked list (insertion, deletion, traversal, reversal). Variations of Linked List (Doubly linked list, circular linked list)
Linked list implementation of Stack and Queue.

Unit-IV: Non- Linear Data Types (15 Lectures)
Trees: Introduction to trees, terminology, Binary Trees, Binary tree representation and traversals (in-order, pre-order, post-order), Binary Search Trees, Applications of trees.
Graphs: Introduction, terminology, linked and matrix representation, Traversing a Graph: BFS, DFS. Dijkstra's Shortest Path Algorithm. Applications of graphs.

TEXT BOOK:

1. Fundamentals of Data Structures in C: Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed.
2. Data Structures & Algorithms; Concepts, Techniques & Algorithms: G. A. V. Pai Tata McGraw Hill.
3. Systematic approach to data structure using C: A M Padma Reddy Sri Nandi Publications

REFERENCES:

1. Mark Allen Weiss, ―Data Structures and Algorithm Analysis in C, Second Edition, Pearson Education, 1996
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, ―Data Structures and Algorithms, Pearson Education, 1983.
3. Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla , ― Data Structures and Program Design in C, Second Edition, Pearson Education, 2007
4. Jean-Paul Tremblay and Paul G. Sorenson, ―An Introduction to Data Structures with Applications, Second Edition, Tata McGraw-Hill, 1991.

5. Pradip Dey and Manas Ghosh, ―Programming in C, Second Edition, Oxford University Press, 2011.
6. Data Structures Through C In Depth S.K.Srivastava, Deepali Srivastava BPB Publications
7. Data Structure Through C Yashavant P. Kanetkar BPB Publications

INTERNSHIP OR PRACTICALS (02 CREDITS: 60 HOURS)

1. Write a C Program to demonstrate the concept of one-dimensional array.
2. Write a C Program to insert and delete element in an array.
3. Write a C Program to Search an element using sequential search
4. Write a C Program to Search an element using binary search
5. Write a C Program to Add two 2-D Matrices
6. Write a C Program to multiply two 2-D Matrices
7. Write a C Program to read a 2-D Matrix and create its equivalent sparse representation.
8. Write a C Program to add two sparse matrices.
9. Write a C Program to implement recursion (factorial)
10. Write a C Program to arrange the list of numbers using Bubble Sort
11. Write a C Program to arrange the list of numbers using Insertion Sort
12. Write a C Program to arrange the list of numbers using Selection Sort.
13. Write a C Program to arrange the list of numbers using quick sort.
14. Write a C Program to implement stack operations using array.
15. Write a C Program to implement basic operations on queue.
16. Write a C Program to implement a circular queue.
17. Write a C Program to convert infix expression to its postfix form using stack operations.
18. Write a C Program to evaluate the given postfix expression using stack operations.
19. Write a C Program to create a singly linked list and perform operations such as insertions and deletions.
20. Write a C Program to reverse a singly linked list.
21. Write a C Program to implement stack and its operations using linked representation.
22. Write a C Program to implement queue and its operations using linked representation.
23. Write a C Program to implement basic operations on doubly linked list.
24. Write a C Program to implement basic operations creation of Binary Search tree.
25. Write a C Program to implement basic operations insertion, search, find min and find max on Binary Search tree.
26. Write a C Program to implement basic operations deletion on Binary Search tree.
27. Write a C Program to implement tree traversal methods
28. Write a C Program to implement breadth first graph traversal
29. Write a C Program to implement Depth first graph traversal.
30. Write a C Program to implement Dijsktra's Shortest Path Algorithm on a graph.

BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-3)
5th SEMESTER

CAP522J3 COMPUTER APPLICATIONS _ DISCRETE MATHEMATICS

CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOMES:

1. *To be able to understand mathematical reasoning in order to read, comprehend, and construct mathematical arguments.*
2. *To be able to count or enumerate objects, and use basic techniques of counting to solve counting problems.*
3. *To be able to work with discrete structures such as sets, permutations, relations, graphs, and trees, and use them to represent discrete objects and the relationships between these objects.*

UNIT 1: (15 HOURS)

Sets, Types, Set Operations, Venn Diagrams, Cardinality, Set Properties and Identities.

Relations, Cartesian Product, Properties of Relations, Types of Relations, Representation of Relations, Closures, Equivalence Relations.

Functions, Domain, Codomain, Range, Types of Functions (one-to-one, many-to-one, onto), Composition of Functions. Algebra of Functions.

UNIT 2: (15 HOURS)

Logic: Propositions, Variables and Logical Operators with Truth Tables. Conditionals and Biconditionals. Converse, Contrapositive, and Inverse of a Proposition.

Truth Tables of Compound Propositions and their equivalence. Tautology, contradiction and contingency. De Morgan's Laws and other Logical Equivalences (laws). Satisfiability.

Predicates. Universal and Existential Quantifiers. Logical Equivalences Involving Quantifiers, Negation and De Morgan's Laws for Quantifiers. Converting English into Logical Expressions. Nested Quantifiers, Negation and Order of Nested Quantifiers. Rules of Inference.

UNIT 3: (15 HOURS)

Introduction to Matrices, operations on matrices (addition, multiplication, transpose). Adjacency Lists, Adjacency Matrices, Incidence Matrices. Counting: Pigeonhole Principle and its Applications.

Permutations, Combinations, Permutations and Combinations with Repetition, Binomial Theorem. Summations, Introduction to Recurrence Relations.

UNIT 4: (15 HOURS)

Graphs, Terminology, Types: Simple, Multigraphs, Directed, Undirected, Complete, Bipartite, Subgraphs. Graph isomorphism. Graph Connectivity, Euler and Hamilton Paths and Circuits, Konigsberg Bridge Problem. Cliques. Graph Coloring: Four Color Theorem.

Trees: Terminology, Arity, Ordered Rooted Trees, Unbalanced and Balanced Trees, Minimal Spanning Trees (Kruskal's & Prim's Algorithm).

DISCRETE MATHEMATICS (TUTORIAL)

1. Find whether $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ is a tautology or a contradiction?
2. Prove the following De-Morgan's law using truth table:
   $\neg (p \wedge q)$ is logically equivalent to $(\neg p \vee \neg q)$
3. State which rule of inference is used in the argument:
   If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have a barbecue tomorrow.
4. Prove by Mathematical Induction that:
   $2 + 6 + 10 + \ldots + (4n - 2) = 2n^2$
5. Give direct proof of the theorem "If n is an odd integer, n^2 is odd."
6. Prove by contraposition that if n is an integer and 3n+2 is odd, then n is odd.
7. Prove that if n=ab, where a and b are positive integers, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.
8. Prove that $\sqrt{2}$ is irrational using proof by contradiction.
9. Let f and g be the functions from the set of integers to a set of integers defined by f(x) =2x+3 and g(x) =3x+2. What is the composition of f and g? What is the composition of g and f?
10. a). Show that $f(x) = x^2+2x+1$ is $O(x^2)$.
    b). Show that n^2 is not O (n).
11. Determine whether the function $f(x) = x^2$ from the set of integers to a set of integers is one-to-one.
12. How many reflexive relations are there on a set with n elements?

13. Let m be a positive integer with m>1. Show that the relation
    a. $R = \{(a,b) | a \equiv b \pmod{m}\}$
    b. is an equivalence relation on a set of integers.
14. Draw the Hasse diagram of (D (75), divides), where the set D (75) represents the set of all divisors of 75. Check whether (D (75), divides) forms a lattice or not.
15. Find the transitive closure for the relation represented by the following matrix:

$$\begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$$

16. How many cards must be selected from a standard deck of 52 cards to guarantee that at least three cards of the same suit are chosen?
17. How many bit strings of length eight either start with a 1 bit or end with two bits 00.
18. Suppose that a department contains 10 men and 15 women. How many ways are there to form a committee with six members if it must have the same number of men and women?
19. A young pair of rabbits (one of each sex) is placed on an island. A pair of rabbits does not breed until they are two months old. After they are two months old, each pair of rabbits produces another pair each month. Find a recurrence relation for the number of pairs of rabbits on the island after n months, assuming that no rabbits ever die.
20. Let H_n denote the number of moves needed to solve the Tower of Hanoi problem with n disks. Set up a recurrence relation for the sequence H_n.
21. What is the solution of the recurrence relation

$$a_n - a_{n-1} + 2a_{n-2}$$
$$\text{with } a_0 = 2 \text{ and } a_1 = 7\ ?$$

22. Find all solutions to the recurrence relation

$$a_n = 5a_{n-1} - 6a_{n-2} + 7^n$$

23. Find all solutions to the recurrence relation

$$a_n = 3a_{n-1} + 2n$$

    What is the solution with $a_1=3$?
24. Draw a graph from following adjacency matrix with respect to ordering of vertices a, b, c, d:

$$\begin{vmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{vmatrix}$$

25. Use Huffman coding to encode the following symbols with the frequencies listed:
    a. A: 0.08 B: 0.10 C: 0.12 D: 0.15 E: 0.20 F: 0.35
    b. What is the average number of bits used to encode the character?
26. Find a binary search tree for the words mathematics physics geography zoology meteorology geology psychology chemistry using alphabetical order.
    Use Prim's algorithm to find a minimum spanning tree of any graph.
27. What is the chromatic number of the complete bipartite graph $K_{m,n}$, where m and n are positive integers?
28. Show that C_6 is bipartite. Also show that K_3 is not bipartite.
29. Use Dijkstra's algorithm to find the length of a shortest path between any two vertices in some weighted connected graph.

TEXTBOOK:

1. Discrete Mathematics, Seymour Lipschutz, marc Lipson, McGraw Hill Publications.
2. Kenneth Rosen, Discrete Mathematics and its Applications (8e, 2019, McGraw-Hill)

REFERENCES:

1. Susanna Epp, Discrete Mathematics with Applications (5e, 2019, Cengage Learning)
2. Eric Lehman, F. Thomson Leighton, Mathematics for Computer Science (1e, 2018)
3. Sarah-Marie Belcastro, Discrete mathematics (2e, 2019, CRC Press)
4. Lipschutz, Schaum's outline of Discrete Mathematics-MGH (2007)
5. GSN Murthy, UM Swami, Mathematics for JEE (Main and Advanced) (Wiley)
6. R S Aggarwal, Senior Secondary School Mathematics for Class XI (Bharati Bhawan)

BACHELOR WITH APPLIED COMPUTING AS MINOR
5th SEMESTER
(FOR STUDENTS WITH MAJOR IN COMPUTER APPLICATIONS / INFORMATION TECHNOLOGY)

ACP522N: APPLIED COMPUTING _ THEORY OF COMPUTATION

CREDITS: THEORY (3) PRACTICAL (1)

MAX. MARKS: 100 MIN. MARKS: 36

COURSE LEARNING OUTCOMES:

Upon completion of this course, students will be able to:

1. *Understand the basic concepts of complexity theory, computability theory, and automata theory.*
2. *Apply different types of proof techniques, such as proof by construction, proof by contradiction, and proof by induction.*
3. *Design and analyze finite automata and regular expressions to recognize and generate regular languages.*
4. *Design and analyze pushdown automata and context-free grammars to recognize and generate context-free languages.*
5. *Understand the Church-Turing thesis and its implications for computability.*
6. *Define and analyze polynomial-time and NP problems.*
7. *Understand the P vs NP problem and its implications for complexity theory.*

THEORY (3 CREDITS)

UNIT 1:

Introduction to Automata Theory: Automata and Languages. Regular Languages: Finite Automata (formal definition of computation, designing finite automata, the regular operations).
Deterministic and non-deterministic automata, formal definition of NFA, equivalence of NFAs and DFAs. Closure under the regular operations.
Regular Expressions: Formal definition, equivalence with finite automata.

UNIT 2:

Introduction to Grammar and its hierarchy. Context Free Languages: Formal definition, examples, designing of CFGs, Ambiguity. Pushdown Automata: Formal definition, equivalence with CFGs.
Non-Context Free Languages. Deterministic Context-Free Languages: properties, Deterministic context-free grammars, relationship of DPDAs and DCFGs.
Recursive and Recursively enumerable languages.

UNIT 3:

Computability Theory. The Church-Turing Thesis: Turing Machines (formal definition, examples). Introduction to Decidability, Reducibility, Recursion Theorem. Introduction to Undecidability: Halting problem.
Complexity Theory. Time Complexity: Asymptotic notations, Class P (polynomial time, examples), The class NP, P vs NP, NP-completeness. Introduction to Space complexity.

TUTORIAL (1 CREDIT):

Examples and problems based on Unit 1st to 3rd as specified by the course teacher.

TEXT BOOK:

1. Introduction to the Theory of Computation, Third Edition Michael Sipser, Cengage Learning, ISBN-13: 978-1-133-18779-0

REFERENCE BOOK

1. Padma Reddy
2. Introduction to Automata Theory, Languages, and Computation by Jeffrey D. Hopcroft, John E. Hopcroft, and Jeffrey D. Ullman (3rd edition, 2006)
3. Formal Languages and Automata Theory by Peter Linz (3rd edition, 2011)

BACHELOR WITH COMPUTER APPLICATIONS AS MINOR (CT-I)
5th SEMESTER

CAP522N: COMPUTER APPLICATIONS _ OPERATING SYSTEM

CREDITS: THEORY (3) PRACTICAL (1)

COURSE LEARNING OUTCOMES:

At the end of this course, a student should be able to:

- *To learn and understand the concepts of operating system.*
- *To understand the core structure and functions of operating system.*
- *Explain process management, processor scheduling, and concurrent programming.*
- *Understand the concept of deadlocks and synchronization.*
- *Learn memory management, file management. I/O systems, and disk scheduling.*
- *Distinguish main memory and virtual memory.*
- *Leam Operating System Design issues.*
- *Leam basic Unix commands and shell programming*

UNIT-1 (15 HOURS)

Introduction: Goals & Structure of Operating System. Basic functions & Modes. Types of Operating System (Batch, Multiprogramming, Multitasking, Time Sharing, Parallel, Distributed A Real-Time OS). System calls.

Process: Process Concept, Process states, Threads, Uni-processor Scheduling: Types of scheduling (Pre-emptive & Non- pre-emptive). Scheduling algorithms (FCFS, SJF, RR, Priority, Thread Scheduling).

UNIT-2 (15 HOURS)

Concurrency: Principles of Concurrency, Mutual Exclusion: Software approaches, Hardware Support, Semaphores, Message Passing, Signals. Monitors, Classical Problems of Synchronization (Readers-Writers, Producer-Consumer, and Dining Philosopher problem).

Deadlock: Principles of Deadlock. Deadlock Prevention & Avoidance, Deadlock Detection.

UNIT-3 (15 HOURS)

Memory: Memory Management requirements, Memory partitioning (Fixed and Variable Partitioning), Memory Allocation Strategies (First Fit, Best Fit and Worst Fit), Fragmentation. Swapping, Segmentation, Paging.

Virtual Memory: Virtual Memory Concepts and Implementation, Concept of Demand Paging, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies). Thrashing.

PRACTICALS (I CREDIT: 30 HOURS)

- In Malling and setting up of Linux /Unix Operating system.
- Using Basic Unix commands like echo, printf. Is, who, date, passwd, cal.
- Using pwd, cd, mkdir, rmdir, cat commands
- Using mv, rm, cp, wc and od commands.
- Demonstrate the use of Pipes.
- Using Basic and Extended regular expressions
- Implementing file permissions, printing commands.
- Mount devices like usb drives in UNIX/Linux.
- Write a shell script to demonstrate the use of variable & basic input output on the console.
- Write a script to demonstrate the use of control statements

TEST BOOK:

- *Peter B Galvin, Greg Gagne. Abraham Silberschatz: Operating System Concepts. John Wiley & Sons, Inc.*
- *Shell Programming by Yashwant Kanetkar*

REFERENCE BOOKS:

- *Peter B Galvin. Greg Gagne. Abraham Silberschatz: Operating System Concepts. John Wiley A Sons, Inc.*
- *Andrew S. Tanenbaum: Modern Operating Systems, PHI*
- *William Stallings: Operating Systems, Pearson Education India*
- *M. G. Venkatesh Murthy: USIX & Shell Programming, Pearson Education*
- *Richard Blum, Christine Bresnahan: Linux Command Line and Shell Scripting Bible, Wiley.*