COURSE OUTCOME

Name of Department: Computer Science

B.A. / B.Sc. / M.A. / M.Sc.	B.Sc. (ECS)	
NAME OF SUBJECT	Operating System	
SEM I / II / III / IV / V / VI	Sem.- III	
COURSE NUMBER (PAPER NUMBER)	**ECS303**	
TITLE OF COURSE (NAME OF PAPER) Operating System		
COURSE CONTENT	OBJECTIVES	OUTCOME
Unit-I: Introduction Operating System: - Definition Operating systems, Types of Operating Systems-Batch, Multiprogramming, Time-Sharing, Real-Time, Distributed, Parallel. OS Service, System components, System Calls.	To learn the fundamentals of Operating Systems and its types.	Analyse the structure of OS and basic architectural components involved in OS design
Unit II: Process Management:- Concept of Process, Process states, Process Control Block, Context switching, Operations on Process, Co-operating Process, Threads – Types of threads, Benefits of threads. Concept of Process Scheduling- Types of Schedulers ,Scheduling criteria , Scheduling algorithms : Preemptive and Non-preemptive, FCFS, SJF, Round Robin, Priority Scheduling, Multilevel Queue Scheduling, Multilevel-feedback Queue Scheduling. Process Synchronization and Deadlocks:-The Producer Consumer Problem, Race Conditions, Critical Section Problem, Semaphores, and Classical Problems of Synchronization: Reader-Writer Problem, Dinning Philosopher Problem, Critical Regions. Definition, System Model, Dead Lock Characterization, Resource Allocation Graph, Methods of Handling Dead Locks- Deadlock Prevention, Deadlock Avoidance -banker's algorithm, resource-request algorithm, Deadlock detection and Recovery.	1. To learn the mechanisms of OS to handle processes and threads and their communication 2. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols	1. Analyse and design the applications to run in parallel either using process or thread models of different OS 2. Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
Unit III: Memory Management:- Basic Hardware Address Binding, Logical and Physical address Space, Dynamic Loading, Overlays, Swapping, and Memory allocation: Contiguous Memory allocation – Fixed and variable partition – Internal and External fragmentation and Compaction, Paging, Segmentation. Basics of Virtual Memory, demand paging, Page fault, Page Replacement policies: Optimal (OPT), First in First Out	To learn the mechanisms involved in memory management in contemporary OS	Analyse the various device and resource management techniques for timesharing and distributed systems

(FIFO), Least Recently used (LRU), Thrashing. Storage Management:- File Management: File concept, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping). Disk Management: disk structure, disk scheduling (FCFS, SSTF, SCAN,CSCAN) , disk reliability, disk Formatting, boot block, bad blocks.		

Signature of HOD

B.A. / B.Sc. / M.A. / M.Sc. B.Sc. (ECS)		
NAME OF SUBJECT Data Structure		
SEM I / II / III / IV / V / VI Sem.- III		
COURSE NUMBER (PAPER NUMBER) **ECS304**		
TITLE OF COURSE (NAME OF PAPER) Data Structure		
COURSE CONTENT	OBJECTIVES	OUTCOME
Unit I: Introduction: Need of Data Structure, Types of Data Structure, ADT, And Algorithm: Definition, characteristics, Space complexity, time complexity, Asymptotic notation (Big O, Omega Ω, theta Φ) Stack: Introduction to stack, Representation-static & dynamic, stack Operations, Application -infix to postfix & prefix, postfix evaluation, recursion, expression validity. Queues: Introduction to Queue, Representation - static & dynamic, Operations, Circular queue, De-queue, priority queues.	To design and implement various data structure algorithms. Introduce the concept of data structures through ADT including Stack and Queues.	Students will be able to implement Linear and Non-Linear data structures. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
Unit II: Linked List:-Introduction to List, Implementation of List – static & dynamic representation, Types of Linked List, Operations on List, Applications of Linked List – polynomial manipulation Trees: Concept & Terminologies, Binary tree, binary search tree, Representation – static &dynamic, Operations on BST – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes	To Understand basic concepts about lists and trees.	Ability to describe linked list and tree operations.
Unit III: Algorithm design strategies: Divide and Conquer, Greedy, Dynamic programming, Backtracking, Branch and Bound. Sorting: Bubble sort, Quick sort, Simple Insertion sort, Shell sort, Address calculation sort, Binary Search Tree, Heap Sort, Merge sort, Radix Sort. Searching: Linear Search, Binary Search, Tree searching methods, Multiway search tree (B-tree, B+ tree), and Height balance tree- AVL trees-Rotations. Hash function (open and	Solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking, and branch and bound and writing programs for these solutions. Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, tournament trees, binary	Select appropriate data structures as applied to specified problem definition. Determine and analyse the complexity of given Algorithms Ability to summarize searching and sorting techniques

close).	search trees, and graphs and writing programs for these solutions.	

Signature of HOD

B.A. / B.Sc. / M.A. / M.Sc. B.Sc. (ECS)		
NAME OF SUBJECT Theory of Computer Science		
SEM I / II / III / IV / V / VI Sem.: - V		
COURSE NUMBER (PAPER NUMBER): - **X**		
TITLE OF COURSE (NAME OF PAPER): - Theory of Computer Science		
COURSE CONTENT	OBJECTIVES	OUTCOME
Unit -I Preliminaries: Basic Definitions, Sets, Various ways of describing a Set, Subsets, operations on Sets, Infinite Sets Relations, Properties of relations, Equivalence of relations.	The student will be able to identify and use set properties and set notation. The student will be able to perform set operations. The student will be able to solve applications involving sets.	Students will demonstrate knowledge of basic mathematical models of set and subset describes how they relate to formal languages.
Unit -II Finite Automata: Introduction, Deterministic Finite Automata, Non Deterministic Finite Automata, The Equivalence of DFA‟s and NFA‟s, Finite Automata with € Moves, Equivalence of NFA with € Transitions and NFA without Transitions, Finite Automata with output , Moore Machine, Melay Machine Equivalence of Moore and Melay Machine.	Understand basic properties of deterministic and nondeterministic finite automata	Define deterministic and nondeterministic finite automata.
Unit-III Regular Expression and Properties of Regular Sets: Regular Expression Operations on set of strings, Regular Expression, Regular Sets, Equivalence of finite automata and regular expression Properties of Regular Sets Closure properties, The pumping lemma of regular sets, Application of pumping lemma.	Be able to prove the equivalence of languages described by finite state machines and regular expressions	Design a finite automaton to recognize a given regular language. Transform a language into regular expression or finite automaton or transition graph. Determine whether a given language is context-free language or not.
Unit-IV Regular and Context Free Grammars: Context Free Grammars (CFG) Derivation and Language generated by grammar, Derivation Trees, Ambiguity of CFG, Simplification of CFG, Normal forms of CFG Regular Grammars Equivalence of regular grammars and finite automata Closure	Understand basic properties of formal languages and formal grammars	Define relationship between regular languages and context-free grammars.

properties of CFG.		
Unit-V Pushdown Automata: Introduction, Definitions, Equivalence of acceptance by final state and empty stack, Definition of DPDA and NPDA their correlation and examples of NPDA, CFG(in GNF) to PDA: Method and example, Closure properties of Regular language, Application of PDA.	Be able to construct pushdown automata and the equivalent context free grammars. Be able to prove the equivalence of languages described by pushdown automata and context free grammars.	Prove properties of context-free languages. Building a context-free grammar for pushdown automata.
Unit-VI Introduction of Turing Machine: Turing Machine model and definition of TM, Language accepted by TM, Design of TM and examples.	Be able to construct Turing machines and Post machines	Design Turing machine and Post machine for a given language. Determine decidability, finiteness and equivalence properties.

Signature of HOD