SCHEME OF EXAMINATION

B. Tech- 2nd Year | **Semester: 4th**

Branch: Computer Science and Business Systems

S.N.	Subject Name	Subject Code	Periods per week			Scheme of Exam			Total Marks	Credit L+(T+P)/2
						Theory/Practical				
			L	T	P	ESE	CT	TA		
1	Design And Analysis Of Algorithms	CS112401	3	1	-	80	20	25	125	3
2	Operating Systems	CS112402	3	1	-	80	20	25	125	3
3	Compiler Design	CS112403	3	1	-	80	20	25	125	3
4	Software Engineering	CS112404	3	1	-	80	20	25	125	3
5	Operations Research	CS112405	3	0	-	80	20	25	125	2
6	Introduction To Innovation, IP Management & Entrepreneurship	CS112406	3	0	-	80	20	25	125	2
7	Design And Analysis Of Algorithms Lab	CS112491		-	2	40	-	20	60	1
8	Operating Systems Lab	CS112492		-	2	40	-	20	60	1
9	Software Engineering Lab	CS112493		-	2	40	-	20	60	1
10	Operations Research Lab	CS112494		-	2	40		20	60	1
	Essence of Indian Traditional Knowledge	CS112496				-	-	10	10	
Total			18	4	8	640	120	240	1000	20

Note:

L-Lecture T-Tutorial P-Practical
CT-Class Test TA-Teachers Assessment ESE-End Semester Exam

Note: **(1)** The teaching in the 1st and 2nd Semester will be divided in two groups consisting of various branches.
(2) Applied Mathematics-I will be taught to both the groups in the first semester.
(3) Self-Learning –I will be conducted by the humanities / relevant discipline as decided by the Director.

Subject Code CS112401	Design And Analysis Of Algorithms	L = 3	T = 1	P = 0	Credits = 3
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	80	20	25	125	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objective	Course Outcomes
The Objective of this course is: • To understand and apply the algorithm analysis techniques. • To critically analyze the efficiency of alternative algorithmic solutions for the same problem. • To understand different algorithm design techniques. • To understand the limitations of Algorithmic power.	**CO1** Design algorithms for various computing problems. **CO2** Analyze the time and space complexity of algorithms. **CO3** Critically analyze the different algorithm design techniques for a given problem. **CO4** Modify existing algorithms to improve efficiency. **CO5** To Understand how to apply various algorithms.

UNIT – I Introduction: Characteristics of Algorithm. Analysis of Algorithm: Asymptotic analysis of Complexity Bounds – Best, Average and Worst-Case behavior; Performance Measurements of Algorithm, Time and Space Trade-Offs, Analysis of Recursive Algorithms through Recurrence Relations: Substitution Method, Recursion Tree Method and Masters' Theorem

UNIT – II Fundamental Algorithmic Strategies: Brute-Force, Heuristics, Greedy, Dynamic Programming, Branch and Bound and Backtracking methodologies; Illustrations of these techniques for Problem-Solving, Bin Packing, Knapsack, Travelling Salesman Problem.

UNIT – III: Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

UNIT – IV: Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.

UNIT – V Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE, Introduction to Quantum Algorithms.

Text Books:

S. No.	Title	Author(s)	Publisher
1	Fundamental of Computer Algorithms,	E. Horowitz and S. Sahni.	

Reference Books:

S. No.	Title	Author(s)	Publisher
1	Introduction to Algorithms,	T. H. Cormen, C. E. Leiserson and R. L. Rivest.	
2	Computer Algorithms:	Introduction to Design and Analysis, S. Baase.	
3	The Art of Computer Programming, Vol. 1, Vol. 2 and Vol. 3, .	D. E. Knuth	

Subject Code CS112402	Operating Systems	L = 3	T = 1	P = 0	Credits = 3
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	80	20	25	125	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objective	Course Outcomes
1. Students will learn how Operating System is Important for Computer System. 2. To make aware of different types of Operating Systemand their services. 3. To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system. 4. To know virtual memoryconcepts. 5. To learn secondary memory Management.	**CO1** To learn what is operating system and how it makes computers work. **CO2** To know how operating system manages complexity through appropriate abstraction of CPU, memory, files, semaphores etc. **CO3** To get knowledge about different components of operating system like Process Management, Concurrencymechanisms. **CO4** To get knowledge about Deadlock handling, Memory Management techniques **CO5** To get knowledge about Virtual Memory, File System

UNIT – I Introduction: Concept of Operating Systems (OS), Generations of OS, Types of OS, OS Services, Interrupt handling and System Calls, Basic architectural concepts of an OS, Concept of Virtual Machine, Resource Manager view, process view and hierarchical view of an OS.

UNIT – II Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
Scheduling algorithms: Pre-emptive and non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.
Inter-process Communication: Concurrent processes, precedence graphs, Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Semaphores, Strict Alternation, Peterson's Solution, The Producer / Consumer Problem, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem, Barber's shop problem.

UNIT – III: Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

Concurrent Programming: Critical region, conditional critical region, monitors, concurrent languages, communicating sequential process (CSP); Deadlocks - prevention, avoidance, detection and recovery.

UNIT – IV: Memory Management: Basic concept, Logical and Physical address maps, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction.
Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page allocation, Partitioning, Paging, Page fault, Working Set, Segmentation, Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

UNIT – V: I/O Hardware: I/O devices, Device controllers, Direct Memory Access, Principles of I/O.
File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed),Free-space management (bit vector, linked list, grouping), directory implementation(linear list, hash table), efficiency and performance.
Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

Case study: UNIX OS file system, shell, filters, shell programming, programming with the standard I/O, UNIX system calls.

Text Books:

S. No.	Title	Author(s)	Publisher
1	Operating System Concepts Essentials	Abraham Silberschatz, Peter Baer	Galvin and Greg Gagne.

Reference Books:

S. No.	Title	Author(s)	Publisher
1	Operating Systems: Internals and Design Principles	William Stallings	
2	Operating System: A Design-oriented Approach.	Charles Patrick Crowley	
3	Operating Systems: A Modern Perspective.	Gary J. Nutt	

Subject Code CS112403	Compiler Design	L = 3	T = 1	P = 0	Credits = 3
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	80	20	25	125	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objectives	Course Outcomes
• To provide the skills needed for building compilers for various situations that one may encounter in a career in Computer Science. • To understanding the fundamental principles in compiler design.	After undergoing the course, students will be able to: 1. Explain the concepts of Compilers and roles of the lexical analyzer. 2. Apply the concepts of different Parsing techniques and implement the knowledge to Yacc tool. 3. Develop syntax directed translation schemes. 4. Implement the principles of scoping, parameter passing and runtime memory management. 5. Use the new code optimization techniques to improve the performance of a program in terms of speed & space and develop algorithms to generate code for a target machine.

UNIT I: Introduction : Introduction to Compiler, single and multi-pass compilers, Translators, Phases of Compilers, Compiler writing tools, Finite Automata and Lexical Analyzer: Role of Lexical Analyzer, Specification of tokens, Recognition of tokens, Regular expression, Finite automata, from regular expression to finite automata, transition diagrams, Implementation of lexical analyzer with LEX.

UNIT II: Syntax Analysis and Parsing Techniques : Context free grammars, Bottom-up parsing and top down parsing, Top down Parsing : elimination of left recursion, recursive descent parsing, Predicative Parsing, Bottom Up Parsing : Operator precedence parsing, LR parsers, Construction of SLR, Canonical LR and LALR parsing tables, Construction of SLR parse tables for Ambiguous grammar, parser generator- YACC, error recovery in top down and bottom up parsing.

UNIT III: Syntax Directed Translation & Intermediate Code Generation : Synthesized and inherited attributes, Construction of syntax trees, bottom up and top down evaluation of attributes, S-attributed and L attributed definitions ,Postfix notation; Three address codes, quadruples, triples and indirect triples, Translation of assignment statements, control flow, Boolean expression and Procedure Calls.

UNIT IV: Run-time Environment : Storage organization, activation trees, activation records, allocation strategies, Parameter passing symbol table, dynamic storage allocation.

UNIT V: Code Optimization and Code Generation : Basic blocks and flow graphs, Optimization of basic blocks, Loop optimization, Global data flow analysis, Loop invariant computations. Issue in the design of Code generator, register allocation, the target machine, and simple Code generator.

Text Books:

S. No.	Title	Author(s)	Publisher
	Compilers Principles, Techniques and Tools	Alfred V. Aho, Ravi Sethi and Ullman J.D	Addison Wesley
	Principle of Compiler Design	Alfred V. Aho and J.D. Ullman	Narosa Publication

Reference Books:

S. No.	Title	Author(s)	Publisher
	Introduction to Compiler Techniques	J.P. Bennet	Tata McGraw-Hill
	Compiler construction (Theory and Practice)	A. Barret William and R.M., Bates	Galgotia Publication
	Compiler Design	O.G. Kakde	Laxmi Publication

Subject Code CS112404	Software Engineering	L=3	T = 1	P = 0	Credits = 3
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	80	20	25	125	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objectives	Course Outcomes
1. To introduce software project and to understand about the different software processes & To introduce ethical and professional issues and to explain why they are concern to software engineers 2. Understanding good coding practices, including documentation, contracts, regression tests and daily builds. 3. Their uses. To understand how Software engineering & Project Management is concerned with theories, methods and tools for professional software development.	After completion of this course, the students would be able to : 1. Developing some basic level of software architecture/design 2. Extracting and analyzing software requirements specifications for different projects 3. Select and implement different software development process models 4. Defining the concepts of software quality and reliability on the basis of International quality standards. 5. Analyzing software risks and risk management strategies 6. Applying different testing and debugging techniques and analyzing their effectiveness. **7.** Defining the basic concepts and importance of Software project management concepts like cost estimation, scheduling and reviewing the progress.

UNIT – I Introduction: Programming in the small vs. programming in the large; software project failures and importance of software quality and timely availability; engineering approach to software development; role of software engineering towards successful execution of large software projects; emergence of software engineering as a discipline.

UNIT – II Software Project Management: Basic concepts of life cycle models – different models and milestones; software project planning – identification of activities and resources; concepts of feasibility study; techniques for estimation of schedule and effort; software cost estimation models and concepts of software engineering economics; techniques of software project control and reporting; introduction to measurement of software size; introduction to the concepts of risk and its mitigation; configuration management

UNIT – III: Software Quality and Reliability: Internal and external qualities; process and product quality; principles to achieve software quality; introduction to different software quality models like McCall, Boehm, FURPS / FURPS+, Dromey, ISO – 9126; introduction to Capability Maturity Models

(CMM and CMMI); introduction to software reliability, reliability models and estimation.

Software Requirements Analysis, Design and Construction: Introduction to Software Requirements Specifications (SRS) and requirement elicitation techniques; techniques for requirement modeling – decision tables, event tables, state transition tables, Petri nets; requirements documentation through use cases; introduction to UML, introduction to software metrics and metrics based control methods; measures of code and design quality.

UNIT – IV: Object Oriented Analysis, Design and Construction: Concepts -- the principles of abstraction, modularity, specification, encapsulation and information hiding; concepts of abstract data type; Class Responsibility Collaborator (CRC) model; quality of design; design measurements; concepts of design patterns; Refactoring; object oriented construction principles; object oriented metrics.

UNIT – V: Software Testing: Introduction to faults and failures; basic testing concepts; concepts of verification and validation; black box and white box tests; white box test coverage – code coverage, condition coverage, branch coverage; basic concepts of black-box tests – equivalence classes, boundary value tests, usage of state tables; testing use cases; transaction based testing; testing for non-functional requirements – volume, performance and efficiency; concepts of inspection.

Text Books:

S. No.	Title	Author(s)	Publisher
1	Software Engineering	Ian Sommerville	

Reference Books:

S. No.	Title	Author(s)	Publisher
1	Fundamentals of Software Engineering,	Carlo Ghezzi, Jazayeri Mehdi	Mandrioli Dino
2	Software Requirements and Specification:	A Lexicon of Practice, Principles and Prejudices,	Michael Jackson
3	The Unified Development Process	Ivar Jacobson, Grady Booch,	James Rumbaugh

Subject Code CS112405	Operations Research	L = 3	T = 0	P = 0	Credits = 2
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	80	20	25	125	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objectives	Course Outcomes
The objective of this course is to impart an understanding the mathematical tools that are needed to solve optimization problems.	On successful completion of the course, the student will be able to: 1. Formulate and solve real-world problems as linear programs for better decision-making. 2. Solve specialized linear programming models like the transportation and assignment Models. 3. Model a dynamic system as a queuing model and compute important performance measures. 4. Use CPM and PERT techniques, to plan, schedule and control project activities. 5. Propose the best strategy using decision making methods under game theory & apply concepts of Simulation to optimize practical problems.

UNIT – I Introduction to OR:

Origin of OR and its definition. Concept of optimizing performance measure, Types of OR problems, Deterministic vs. Stochastic optimization, Phases of OR problem approach – problem formulation, building mathematical model, deriving solutions, validating model, controlling and implementing solution.

UNIT – II Linear Programming:

Linear programming – Examples from industrial cases, formulation & definitions, Matrix form. Implicit assumptions of LPP. Some basic concepts and results of linear algebra – Vectors, Matrices, Linear Independence/Dependence of vectors, Rank, Basis, System of linear eqns., Hyperplane, Convex set, Convex polyhedron, Extreme points, Basic feasible solutions. Geometric method: 2-variable case, Special cases – infeasibility, unboundedness, redundancy &degeneracy, Sensitivity analysis. Simplex Algorithm – slack, surplus & artificial variables, computational details, big-M method, identification and resolution of special cases through simplex iterations. Duality – formulation, results, fundamental theorem of duality, dual-simplex and primal-dual algorithms

UNIT – III: Transportation and Assignment problems:

TP - Examples, Definitions – decision variables, supply & demand constraints, formulation, Balanced & unbalanced situations, Solution methods – NWCR, minimum cost and VAM, test for optimality(MODI method), degeneracy and its resolution. AP - Examples, Definitions – decision variables, constraints, formulation, Balanced &unbalanced situations, Solution method – Hungarian, test for optimality (MODI method), degeneracy & its resolution.

UNIT – IV PERT – CPM:

Project definition, Project scheduling techniques – Gantt chart, PERT & CPM, Determination of

critical paths, Estimation of Project time and its variance in PERT using statistical principles, Concept of project crashing/time-cost trade-off.

UNIT – V: Inventory Control:
Functions of inventory and its disadvantages, ABC analysis, Concept of inventory costs, Basics of inventory policy (order, lead time, types), Fixed order-quantity models – EOQ, POQ & Quantity discount models. EOQ models for discrete units, sensitivity analysis and Robustness, Special cases of EOQ models for safety stock with known/unknown stock out situations, models under prescribed policy, Probabilistic situations.

Text Books:

S. No.	Title	Author(s)	Publisher
1	Operations Research:	An Introduction. H.A. Taha.	

Reference Books:

S. No.	Title	Author(s)	Publisher
1	Linear Programming.	K.G. Murthy.	
2	Linear Programming.	G. Hadley.	
3	Principles of OR with Application to Managerial Decisions.	H.M. Wagner.	

Subject Code CS112406	Introduction To Innovation, IP Management & Entrepreneurship	L = 3	T = 0	P = 0	Credits = 2
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	80	20	25	125	3 Hours
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objectives	Course Outcomes

UNIT – I Innovation: What and Why

Innovation as a core business process, Sources of innovation, Knowledge push vs. need pull innovations. Class Discussion- Is innovation manageable or just a random gambling activity?

UNIT – II Building an Innovative Organization

Creating new products and services, Exploiting open innovation and collaboration, Use of innovation for starting a new venture Class Discussion- Innovation: Co-operating across networks vs. 'go-it-alone' approach

UNIT – III Entrepreneurship:

- Opportunity recognition and entry strategies
- Entrepreneurship as a Style of Management
- Maintaining Competitive Advantage- Use of IPR to protect Innovation

UNIT – IV Entrepreneurship- Financial Planning:

- Financial Projections and Valuation
- Stages of financing
- Debt, Venture Capital and other forms of Financing

UNIT – V Intellectual Property Rights (IPR)

- Introduction and the economics behind development of IPR: Business Perspective
- IPR in India – Genesis and Development

- International Context
- Concept of IP Management, Use in marketing

Types of Intellectual Property

- Patent- Procedure, Licensing and Assignment, Infringement and Penalty
- Trademark- Use in marketing, example of trademarks- Domain name
- Geographical Indications- What is GI, Why protect them?
- Copyright- What is copyright

Industrial Designs- What is design? How to protect?

Text Books:

S. No.	Title	Author(s)	Publisher
1	Managing Innovation: Integrating Technological, Market and Organizational Change	Joe Tidd, John Bessant.	
2	Case Study Materials: To be distributed for class discussion		

Subject Code CS112491	Design And Analysis Of Algorithms Lab	L =	T = 0	P = 02	Credits = 1
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	40	-	20	60	-
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Experiments

1. Sort a given set of elements using the Bubble sort method and determine the time required to sort the elements.
2. Sort a given set of elements using the Quick sort method and determine the time required to sort the elements.
3. Implement a Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements.
4. Implement a Insertion Sort algorithm to sort a given set of elements and determine the time required to sort the elements.
5. Implement recursive binary search and determine the time required to search an element.
6. Implement Linear search and determine the time required to search an element.
7. Check whether a given graph is connected or not using DFS method.
8. Print all the nodes reachable from a given starting node in a digraph using BFS method.
9. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
10. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
11. Implement N Queen's problem using Back Tracking.
12. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

Subject Code CS112492	Operating Systems Lab (Unix Environment)	L =	T = 0	P = 02	Credits = 1
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	40	-	20	60	-
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Course Objective	Course Outcomes
The objectives of this lab is : 1. To familiarize students with the architecture of Unix OS. 2. To provide necessary skills for developing and debugging programs in UNIX environment.	**CO1** Students will be able to Work confidently in Unix/Linux environment **CO2** Students will able to Write shell scripts to automate various tasks **CO3** Students will understand different commands on Unix and its usage **CO4** Students will be able to describe and use thefundamental UNIX system tools and utilities **CO5** Students will understand different kernel algorithmsand its **usage**

1. Unix commands (files directory, data manipulation, network communication etc), shell programming and vi editor
2. C program implementation of the following:
   a. Scheduling Algorithms
   b. Shared memory
   c. Thread and Multi Thread
   d. Inter Process Communication
   e. Deadlock Avoidance and Deadlock Detection
   f. Semaphore
   g. Memory Management
   h. Indexing and Hashing

Text Books:

1. Operating System Concepts Essentials. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne.

Reference Books:

1. Operating Systems: Internals and Design Principles. William Stallings.
2. Operating System: A Design-oriented Approach. Charles Patrick Crowley.
3. Operating Systems: A Modern Perspective. Gary J. Nutt.
4. Design of the Unix Operating Systems. Maurice J. Bach.
5. Understanding the Linux Kernel, Daniel Pierre Bovet, rco CesatiMa.

Subject Code CS112493	Software Engineering Lab	L =	T = 0	P = 2	Credits = 1
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	40	-	20	60	-
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

Laboratory

- Development of requirements specification, function oriented design using SA/SD,
- Object-oriented design using UML, test case design, implementation using C++ and testing.
- Use of appropriate CASE tools and other tools such as configuration management tools, program analysis tools in the software life cycle.

Subject Code CS112494	Operations Research Lab	L =	T = 0	P = 02	Credits = 1
Examination Scheme	ESE	CT	TA	Total	ESE Duration
	40	-	20	60	-
	Minimum number of class tests to be conducted=02			Minimum Assignments=02	

1. Formulation of linear programming problems.
2. Solution of linear programming problem using graphical method with:
   i. Multiple constraints
   ii. Unbounded solution
   iii. Infeasible solution
   iv. Alternative or multiple solution
3. Enumeration of all basic solutions for linear programming problem.
4. Solution of linear programming problem with simplex method.
5. Problem solving using Big M method.
6. Problem solving using two phase method.
7. Solution on primal problem as well as dual problem.
8. Solution based on dual simplex method.
9. Verification of weak duality, strong duality and complementary slackness property.
10. Solution of transportation problem.
11. Solution of assignment problem.
12. Solution of integer programming problem using Branch and Bound method.
13. Solution of integer programming problem using Gomory's cutting plane method.
14. Simulation: Random number generation.
15. Monte Carlo method.
16. Performance measures for M/M/1 queuing model.
17. ABC analysis.
18. Inventory model.