Bachelor of Technology (Computer Science & Engineering)  
Scheme of Studies/Examination  
Semester III

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course No.</th>
<th>Subject</th>
<th>L:T:P</th>
<th>Hours/Week</th>
<th>Examination Schedule (Marks)</th>
<th>Duration of Exam (Hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Major Test</td>
<td>Minor Test</td>
</tr>
<tr>
<td>1</td>
<td>HS-201 N</td>
<td>Fundamentals of Management</td>
<td>3:0:0</td>
<td>3</td>
<td>75 25 0</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>CSE-201 N</td>
<td>Discrete Structures</td>
<td>3:1:0</td>
<td>4</td>
<td>75 25 0</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>CSE-203 N</td>
<td>Data Structures</td>
<td>3:1:0</td>
<td>4</td>
<td>75 25 0</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>CSE-205 N</td>
<td>Data Base Management Systems</td>
<td>3:1:0</td>
<td>4</td>
<td>75 25 0</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>CSE-207 N</td>
<td>Digital Electronics</td>
<td>3:1:0</td>
<td>4</td>
<td>75 25 0</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>CSE-209 N</td>
<td>Programming Languages</td>
<td>3:1:0</td>
<td>3</td>
<td>75 25 0</td>
<td>100</td>
</tr>
<tr>
<td>7</td>
<td>CSE-211 N</td>
<td>Data Structures Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0 40 60</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>CSE-213 N</td>
<td>Digital Electronics Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0 40 60</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>CSE-215 N</td>
<td>Database Management Systems Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0 40 60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td>31</td>
<td>450 270 180</td>
<td>900</td>
</tr>
<tr>
<td>10</td>
<td>MPC 202 N</td>
<td>Energy Studies*</td>
<td>3:0:0</td>
<td>3</td>
<td>75 25 0</td>
<td>100</td>
</tr>
</tbody>
</table>

*MPC-202 is a mandatory course which will be a non credit subject and student has to get pass marks in order to qualify for the Degree award.
Purpose: To make the students conversant with the basics concepts in management thereby leading to nurturing their managerial skills

COURSE OUTCOMES

CO1 An overview about management as a discipline and its evolution
CO2 Understand the concept and importance of planning and organizing in an organization
CO3 Enabling the students to know about the importance of hiring and guiding the workforce by understanding the concept of leadership and communication in detail
CO4 To understand the concept and techniques of controlling and new trends in management

UNIT-1
Introduction to Management: Meaning, Definition, nature, importance & Functions, Management as Art, Science & Profession - Management as social System, Concepts of management-Administration

UNIT-II
Planning: nature, purpose and functions, types of plans, planning process, Strategies and Policies: Concept of Corporate Strategy, formulation of strategy, Types of strategies, Management by objectives (MBO), SWOT analysis, Types of policies, principles of formulation of policies
4. Organizing: nature, importance, process, organization structure: Line and Staff organization, Delegation of Authority and responsibility, Centralization and Decentralization, Decision Making Process, Decision Making Models, Departmentalization: Concept and Types (Project and Matrix), formal & informal organizations

UNIT-III
Staffing: concept, process, features; manpower planning; Job Analysis: concept and process; Recruitment and selection: concept, process, sources of recruitment; performance appraisal, training and development
Directing: Communication- nature, process, formal and informal, barriers to Effective Communication, Theories of motivation-Maslow, Herzberg, Mc Gregor; Leadership – concept and theories, Managerial Grid, Situational Leadership. Transactional and Transformational Leadership

UNIT-IV
Controlling: concept, process, types, barriers to controlling, controlling Techniques: budgetary control, Return on investment, Management information system-MIS, TQM-Total Quality Management, Network Analysis-PERT and CPM.
Recent Trends in Management: Social Responsibility of Management–Management of Crisis, Total Quality Management, Stress Management, Concept of Corporate Social Responsibility (CSR) and business ethics.
Functional aspects of business: Conceptual framework of functional areas of management- Finance; Marketing and Human Resources

Text books
1. Management Concepts - Robbins, S.P; Pearson Education India

Recommended books
2. Management and OB– Mullins; Pearson Education
4. Management Theory and Practice – Gupta, C.B; Sultan Chand and Sons, new Delhi
7. Organizational behavior – Robins Stephen P; PHI.
Unit 1 Set Theory & Logic

Unit 2: Relations, digraphs and lattices
Product sets and partitions, relations and digraphs, paths in relations and digraphs, properties of relations, equivalence and partially ordered relations, computer representation of relations and digraphs, manipulation of relations, Transitive closure and Warshall’s algorithm, Posets and Hasse Diagrams, Lattice.

Unit 3 Functions and Combinatorics
Definitions and types of functions: injective, subjective and bijective, Composition, identity and inverse, Review of Permutation and combination-Mathematical Induction, Pigeon hole principle, Principle of inclusion and exclusion, Generating function-Recurrence relations.

Unit 4: Algebraic Structures
Algebraic structures with one binary operation - semi groups, monoids and groups, Product and quotient of algebraic structures, Isomorphism, homomorphism, automorphism, Cyclic groups, Normal sub group, codes and group codes, Ring homomorphism and Isomorphism.

Books:
- Discrete mathematical structures by B Kolman RC Busby, S Ross PHI Pvt. Ltd.
- Discrete Mathematics by Bisht & Dhami, Oxford University Press, 2015

Reference:
- Discrete Mathematics for computer scientists and Mathematicians, Joe L. Mott, Abraham
Course Outcomes (CO)

CO 1 To introduce the basic concepts of Data structure, basic data types, searching and sorting based on array data types.
CO 2 To introduce the structured data types like Stacks, Queue, and its basic operations' implementation.
CO 3 To introduces dynamic implementation of linked list.
CO 4 To introduce the concepts of Tree and graph and implementation of traversal algorithms.

Unit-1

Introduction to Data Structures, Data Types, Built in and User Defined Data Structures, Applications of Data Structure, Algorithm Analysis, Worst, Best and Average Case Analysis, Notations of Space and Time Complexity, Arrays, One Dimensional Arrays, Two Dimensional Arrays and Multi-Dimensional Arrays, Sparse Matrices, Storage Class, Basics of Recursion.

Searching from array using Linear and Binary Searching Algorithm, Sorting of array using Selection, Insertion, Bubble, Radix Algorithm

Unit-2


Queues: Definition, Sequential Implementation of Linear Queues and Its Operations, Circular Queue and Its Implementation, Priority Queues and Its Implementation, Applications of queues.

Unit-3


Unit-4

Trees: Definition, Basic Terminology, Binary Tree, External and Internal Nodes, Static and Dynamic Implementation of a Binary Tree, Primitive Operations on Binary Trees, Binary Tree Traversals: Per-Order, In-Order And Post-Order Traversals. Representation of Infix, Post-Fix and Prefix Expressions using Trees.
Graphs: Basic Terminology, Definition of Undirected & Directed Graphs, Memory Representation of Graphs, Minimum-Spanning Trees, Warshal Algorithm, Graph Traversals Algorithms: Breadth First and Depth First,

Text Book:
- Theory & Problems of Data Structures by Jr. Symour Lipschetz, Schaum’s outline by TMH
- Fundamentals of Data structures by Ellis Horowitz & Sartaj Sahni, Pub, 1983,AW
- Data Structures Using C,2E by Reema Thareja,Oxford University Press,2014

References:
- Shukla, Data Structures using C++, Wiley India
- Fundamentals of computer algorithms by Horowitz Sahni and Rajasekaran.
- Data Structures and Program Design in C by Robert Kruse, PHI,
- Shukla, Data Structures using C++, Wiley India
UNIT I

Introduction: Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS.


UNIT II

The Relational Data Model & Algebra

Relational Model: Structure of relational Databases, Relational Algebra, Relational Calculus, introduction to Views, updates on views

SQL and Integrity Constraints: Concept of DDL, DML, DCL. Basic Structure, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Sub queries, Database security application development using SQL, Stored procedures and triggers.

UNIT III

Relational Database Design:

Functional Dependency, Different anomalies in designing a Database, Normalization using functional dependencies, Decomposition, Boyce-Codd Normal Form, 3NF, Normalization using multi-valued dependencies, 4NF, 5NF

Internals of RDBMS:

Physical data structures, Query optimization: join algorithm, statistics and cost base optimization. Transaction processing, Concurrency control and Recovery Management: transaction model properties, state serializability, lock base protocols, two phase locking.

UNIT IV

Failure Recovery and Concurrency Control


Concurrency Control: Serial and Serializable Schedules—Conflict Serializability—Enforcing Serializability by Locks—Locking Systems with Several Lock Modes—Concurrency Control by Timestamps, validation.

Transaction Management: Serializability and Recoverability—View, Serializability—Resolving Deadlocks—Distributed Databases: Commit and Lock

Text Books:
1. Korth, Silberschatz, Sudarshan: database concepts, MGH,

Reference Books:
1. R. Ramakrishnan and J. Gehrks database management system; MGH, International edition,
2. C. J. Date, data base systems: 7th edition, Addison Wesley, Pearson Education,
Chakrabarti, Advance database management systems, Wiley Dreamtech
Purpose
To learn the basic methods for the design of digital circuits and provide the fundamental concepts used in the design of digital systems.

Course Outcomes

CO 1 To introduce basic postulates of Boolean algebra and shows the correlation between Boolean expressions

CO 2 To introduce the methods for simplifying Boolean expressions

CO 3 To outline the formal procedures for the analysis and design of combinational circuits and sequential circuits

CO 4 To introduce the concept of memories and programmable logic devices.

UNIT I MINIMIZATION TECHNIQUES AND LOGIC GATES
Minimization Techniques: Boolean postulates and laws - De-Morgan's Theorem, Principle of Duality, Boolean expression - Minimization of Boolean expressions, Minterm, Maxterm, Sum of Products (SOP), Product of Sums (POS), Karnaugh map Minimization - Don't care conditions, Quine - McCluskey method of minimization. Logic Gates: AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR- Implementations of Logic Functions using gates, NAND-NOR implementations - Multi level gate implementations- Multi output gate implementations. TTL and CMOS Logic and their characteristics, Tristate gates.

UNIT II COMBINATIONAL CIRCUITS

UNIT III SEQUENTIAL CIRCUITS

UNIT IV MEMORY DEVICES
Classification of memories - ROM: ROM organization, PROM, EPROM, EEPROM, EAPROM, RAM: - RAM organization - Write operation, Read operation, Memory cycle, Timing wave forms, Memory decoding, memory expansion, Static RAM Cell, Bipolar RAM cell structure, Dynamic RAM cell structure, Programmable Logic Devices - Programmable Logic Array (PLA), Programmable Array Logic (PAL), Implementation of PLA, PAL using ROM. Introduction to Field Programmable Gate Arrays (FPGA).

TEXT BOOKS

REFERENCES
1. A.K. Maini, Digital Electronics, Wiley India
### Course Information

**CSE-209 N**

#### Programming Languages

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3 Hour</td>
</tr>
</tbody>
</table>

**Purpose**

To introduce the principles and paradigms of programming languages for design and implement the software intensive systems.

<table>
<thead>
<tr>
<th>Course Outcomes (CO)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CO 1</strong></td>
</tr>
<tr>
<td><strong>CO 2</strong></td>
</tr>
<tr>
<td><strong>CO 3</strong></td>
</tr>
<tr>
<td><strong>CO 4</strong></td>
</tr>
</tbody>
</table>

### Unit-I: Introduction, Syntax and Semantics

**Introduction:** A brief history, Characteristics of a good programming language, Programming language translators compiler & interpreters , Elementary data types – data objects, variable & constants, data types, Specification & implementation of elementary data types, Declarations, type checking & type conversions, Assignment & initialization, Numeric data types, enumerations, Booleans & characters.

**Syntax & Semantics:** Introduction, general problem of describing syntax, formal method of describing syntax, attribute grammar dynamic semantic.

### Unit-II: Structured data objects, Subprograms and Programmer Defined Data Types

**Structured data objects:** Structured data objects & data types, specification & implementation of structured data types, Declaration & type checking of data structure, vector & arrays, records Character strings, variable size data structures, Union, pointer & programmer defined data objects, sets, files.

**Subprograms and Programmer Defined Data Types:** Evolution of data type concept abstraction, encapsulation & information hiding, Subprograms, type definitions, abstract data types, over loaded subprograms, generic subprograms.

### Unit–III: Sequence Control and Data Control

**Sequence Control:** Implicit & explicit sequence control, sequence control within expressions, sequence control within statement, Subprogram sequence control: simple call return, recursive subprograms, Exception & exception handlers, co routines, sequence control. Concurrency – subprogram level concurrency, synchronization through semaphores, monitors & message passing

**Data Control:** Names & referencing environment, static & dynamic scope, block structure, Local data & local referencing environment, Shared data: dynamic & static scope, Parameter & parameter transmission schemes.

### Unit-IV: Storage Management and Programming Languages

**Storage Management:** Major run time elements requiring storage, programmer and system controlled storage management & phases, Static storage management, Stack based storage management, Heap storage management, variable & fixed size elements.

**Programming Languages:** Introduction to procedural, non-procedural, structured, logical, functional and object oriented programming language, Comparison of C & C++ programming languages.

### Text Books:

1. Terrence W. Pratt, Marvin V. Zelkowitz, Programming Languages Design & Implementation, Pearson.

### Reference Books:

CSE-211 N  
Data Structures Lab

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Minor Test</th>
<th>Practical</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

**Purpose**  
To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically

**Course Outcomes (CO)**

**CO 1**  
To introduce the basic concepts of Data structure, basic data types, searching and sorting based on array data types.

**CO 2**  
To introduce the structured data types like Stacks and Queue and its basic operation’s implementation.

**CO 3**  
To introduces dynamic implementation of linked list.

**CO 4**  
To introduce the concepts of Tree and graph and implementation of traversal algorithms.

1. Write a program for Binary search methods.
2. Write a program for insertion sort, selection sort and bubble sort.
3. Write a program to implement Stack and its operation.
4. Write a program for quick sort.
5. Write a program for merge sort.
6. Write a program to implement Queue and its operation.
7. Write a program to implement Circular Queue and its operation.
8. Write a program to implement singly linked list for the following operations: Create, Display, searching, traversing and deletion.
9. Write a program to implement doubly linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
10. Write a program to implement circular linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.
11. Write a program to implement insertion, deletion and traversing in B tree

**NOTE:**
At least seven experiments are to be performed from above list and the concerned institution as per the scope of the syllabus may set remaining eight.
CSE-213 N
Digital Electronics Lab

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Minor Test</th>
<th>Practical</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

Purpose: To learn the basic methods for the design of digital circuits and systems.

Course Outcomes

<table>
<thead>
<tr>
<th>CO 1</th>
<th>To Familiarization with Digital Trainer Kit and associated equipment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO 2</td>
<td>To Study and design of TTL gates</td>
</tr>
<tr>
<td>CO 3</td>
<td>To learn the formal procedures for the analysis and design of combinational circuits.</td>
</tr>
<tr>
<td>CO 4</td>
<td>To learn the formal procedures for the analysis and design of sequential circuits</td>
</tr>
</tbody>
</table>

LIST OF EXPERIMENTS:

1. Familiarization with Digital Trainer Kit and associated equipment.
2. Study of TTL gates AND, OR, NOT, NAND, NOR, EX-OR, EX-NOR.
4. To verify the operation of Multiplexer and De-multiplexer.
5. To verify the operation of Comparator.
7. To verify the operation of Bi-directional shift register.
8. To design and verify the operation of 3-bit asynchronous counter.
9. To design and verify the operation of asynchronous Up/down counter using J-K FFs.
10. To design and verify the operation of asynchronous Decade counter.
11. Study of TTL logic family characteristics.
13. Study of BCD to 7 segment Decoder.

NOTE:
At least ten experiments are to be performed from above list and the concerned institution as per the scope of the syllabus may set remaining five.
1. Write the queries for Data Definition Language (DDL) in RDBMS.
2. Write the queries for Data Manipulation Language (DML) in RDBMS.
3. Write the queries for Data Control Language (DCL) in RDBMS.
4. Write SQL queries using logical operations (=, etc)
5. Write SQL queries using SQL operators
6. Write SQL query using character, number, date and group functions
7. Write SQL queries for relational algebra
8. Write SQL queries for extracting data from more than one table
9. Write SQL queries for sub queries, nested queries
10. Concepts for ROLL BACK, COMMIT & CHECK POINTS
11. Create VIEWS, CURSORS and TR
12. High level language extension with Cursors.
13. High level language extension with Triggers.
14. To study the concept of Procedures and Functions.
MPC-202 N

ENERGY STUDIES

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

**Purpose**: To make the students conversant with the basics concepts and conversion of various form of Energy

**Course Outcomes**

<table>
<thead>
<tr>
<th>CO1</th>
<th>An overview about Energy, Energy Management, Audit and tariffs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>Understand the Layout and working of Conventional Power Plants</td>
</tr>
<tr>
<td>CO3</td>
<td>Understand the Layout and working of Non-Conventional Power Plants</td>
</tr>
<tr>
<td>CO4</td>
<td>To understand the role of Energy in Economic development and Energy Scenario in India</td>
</tr>
</tbody>
</table>

**Introduction**: Types of energy, Conversion of various forms of energy, Conventional and Non-conventional sources, Need for Non-Conventional Energy based power generation.


**Energy Audit**: Need, Types, Methodology and Approach.

**UNIT-II**

**Conventional Energy sources**: Selection of site, working of Thermal, Hydro, Nuclear and Diesel power plants and their schematic diagrams & their comparative advantages - disadvantages.

**UNIT-III**

**Non-Conventional Energy sources**: Basic principle, site selection of Solar energy power plant, photovoltaic technologies, PV Systems and their components, Wind energy power plant, Bio energy plants, Geothermal energy plants and tidal energy plants. MHD

**UNIT-IV**

**Energy Scenario**: Lay out of power system, Role of Energy in Economic development, energy demand, availability and consumption, Commercial and Non-commercial energy, Indian energy scenario, long term energy scenario, energy pricing, energy sector reforms in India, energy strategy for the future.

**References**:

1. Energy Studies-Wiley Dream tech India.
4. NEDCAP: Non Conventional Energy Guide Lines
5. G. D. Roy: Non conventional energy sources
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course No.</th>
<th>Subject</th>
<th>L:T:P</th>
<th>Hours/Week</th>
<th>Examination Schedule (Marks)</th>
<th>Duration of Exam (Hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Major Test</td>
<td>Minor Test</td>
</tr>
<tr>
<td>1</td>
<td>AS-201 N</td>
<td>Mathematics-III</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>CSE-202 N</td>
<td>Object Oriented Programming</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>CSE-204 N</td>
<td>Internet Fundamental</td>
<td>3:0:0</td>
<td>3</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>CSE-206 N</td>
<td>Digital Data Communication</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>CSE-208 N</td>
<td>Microprocessor &amp; Interfacing</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>CSE-210 N</td>
<td>Operating System</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>CSE-212 N</td>
<td>Object Oriented Programming Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>CSE-214 N</td>
<td>Microprocessor Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>CSE-216 N</td>
<td>Internet Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>MPC 201 N</td>
<td>Environment Studies*</td>
<td>3:0:0</td>
<td>3</td>
<td>75</td>
<td>25</td>
</tr>
</tbody>
</table>

Total: 32 450 270 180 900

*MPC-201 is a mandatory course which will be a non credit subject and student has to get pass marks in order to qualify for the Degree award*
AS-201 N  
Mathematics-III

<table>
<thead>
<tr>
<th>Purpose</th>
<th>To provide the conceptual knowledge of Engineering mathematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3</td>
</tr>
<tr>
<td>Tutorial</td>
<td>1</td>
</tr>
<tr>
<td>Practical</td>
<td>-</td>
</tr>
<tr>
<td>Major Test</td>
<td>75</td>
</tr>
<tr>
<td>Minor Test</td>
<td>25</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
<tr>
<td>Time</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO 1</td>
</tr>
<tr>
<td>CO 2</td>
</tr>
<tr>
<td>CO 3</td>
</tr>
<tr>
<td>CO 4</td>
</tr>
</tbody>
</table>

UNIT – I
Fourier series: Euler’s Formulae, Conditions for Fourier expansions, Fourier expansion of functions having points of discontinuity, change of interval, Odd & even functions, Half-range series.

UNIT-II
Functions of Complex Variables: Functions of a complex variable, Exponential function, Trigonometric, Hyperbolic and Logarithmic functions, limit and continuity of a function, Differentiability and analyticity. Cauchy-Riemann equations, Necessary and sufficient conditions for a function to be analytic, Polar form of the Cauchy-Riemann equations, Harmonic functions, Application to flow problems, Conformal transformation, Standard transformations (Translation, Magnification & rotation, inversion & reflection, Bilinear).

UNIT-III
Probability Distributions: Probability, Baye’s theorem, Discrete & Continuous probability distributions, Moment generating function, Probability generating function, Properties and applications of Binomial, Poisson and normal distributions.

UNIT-IV
Linear Programming: Linear programming problems formulation, Solution of Linear Programming Problem using Graphical method, Simplex Method, Dual-Simplex Method.

Text Book

Reference Book
3. Operation Research: H.A. Taha
4. Probability and statistics for Engineer: Johnson. PHI.
<table>
<thead>
<tr>
<th>CSE-202 N</th>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Oriented Programming</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3 Hour</td>
</tr>
</tbody>
</table>

**Purpose**
To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System

**Course Outcomes (CO)**

| CO 1 | To introduce the basic concepts of object oriented programming language and its representation |
| CO 2 | To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation. |
| CO 3 | To introduce polymorphism, interface design and overloading of operator. |
| CO 4 | To handle backup system using file, general purpose template and handling of raised exception during programming |

**Unit-1**

**Unit-2**
Friend Function and Friend Classes, This Pointer, Dynamic Memory Allocation and Deallocation (New and Delete), Static Class Members, Constructors, parameter Constructors and Copy Constructors, Deconstructors, Introduction of inheritance, Types of Inheritance, Overriding Base Class Members in a Derived Class, Public, Protected and Private Inheritance, Effect of Constructors and Deconstructors of Base Class in Derived Classes.

**Unit-3**

**Unit-4**

**Text Books:**
- The complete reference C++ by Herbert shieldt Tata McGraw Hill

**References Books**
- Shukla, Object Oriented Programming in c++, wiley india
- Programming with C++ By D Ravichandran, 2003, T.M.H
UNIT-1 : THE INTERNET
Introduction to networks and internet, history, Internet, Intranet & Extranet, Working of Internet, Internet Congestion, internet culture, business culture on internet. Collaborative computing & the internet. Modes of Connecting to Internet, Internet Service Providers(ISPs), Internet address, standard address, domain name, DNS, IP:v6. Modems, Speed and time continuum, communications software; internet tools.

UNIT-II : WORLD WIDW WEB

UNIT-III : INTERNET PLATFORM AND MAILING SYSTEMS
Introduction, advantages and disadvantages, User Ids, Pass words, e-mail addresses, message components, message composition, mailer features, E-mail inner workings, E-mail management, MIME types, Newsgroups, mailing lists, chat rooms, secure-mails, SMTP, PICO, Pine, Library cards catalog, online ref. works.
Languages: Basic and advanced HTML, Basics of scripting languages – XML, DHTML, Java Script.

UNIT-IV : SERVERS
Introduction to Web Servers: PWS, IIS, Apache; Microsoft Personal Web Server. Accessing & using these servers.
Privacy and security topics: Introduction, Software Complexity, Attacks, security and privacy levels, security policy, accessibility and risk analysis, Encryption schemes, Secure Web document, Digital Signatures, Firewalls, Intrusion detection systems

Text Book:
• Internet & World Wide Programming, Deitel,Deitel & Nieto, 2012, Pearson Education
• Fundamentals of the Internet and the World Wide Web, Raymond Greenlaw and Ellen Hepp, TMH- 2012

Reference Books:
• Complete idiots guide to java script., Aron Weiss, QUE, 2013
• Network firewalls, Kironjeet syan -New Rider Pub.2014
• www.secinf.com
• www.hackers.com
• Alfred Glkossbrenner-Internet 101 Computing MGH, 2013
Digital Data Communication

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

**Purpose**
To provide the conceptual knowledge of data preparation and signal transmission methodologies used in data communication and networking.

**Course Outcomes**

<table>
<thead>
<tr>
<th>CO 1</th>
<th>To study various analog communication techniques and with their characteristics.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO 2</td>
<td>To study and understand the requirements for analog/digital data to analog/digital signal conversion techniques.</td>
</tr>
<tr>
<td>CO 3</td>
<td>To study the error and flow control techniques in communication and networking.</td>
</tr>
<tr>
<td>CO 4</td>
<td>To study the concept of multiplexing and applied multiple access techniques specially in satellite communication.</td>
</tr>
</tbody>
</table>

**UNIT-I**

**MODULATION TECHNIQUES**
Basic constituents of Communication Systems need of modulation, Amplitude modulation, spectrum of AM wave, modulation index, DSBSC modulation, SSB Modulation, vestigial side band modulation.

**ANGLE MODULATION:** Frequency and Phase Modulation, spectrum of FM Wave, modulation index and Bandwidth of FM Signal, NBFM and WBFM.

**UNIT-II**

**DATA ENCODING**

**UNIT-III**

**DIGITAL DATA COMMUNICATION TECHNIQUES**


**UNIT-IV**

**SATELITE COMMUNICATION**
Multiplexing: Advantages – Types of Multiplexing – FDM – Synchronous TDM – Statistical TDM or Asynchronous TDM, Study of their characteristics.

**Satellite Communication Systems:** Satellite parameters and configurations – Capacity allocation, Frequency Division FDMA, Time Division TDMA- Fixed assigned multiple access (FAMA), Demand assign multiple access (DAMA) – The concept of spread spectrum: FHSS, DSSS – CDMA – Transmission and reception.

**TEXT BOOKS**

**REFERENCES**
1. Stallings, “Data & computer Communications”,PHI.
3. Irvine, Data communications & Networks An engineering approach, wiley india
CSE-208 N  
Microprocessor & Interfacing

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

Purpose  
To learn the architecture and programming of Intel family microprocessors and its interfacing.

Course Outcomes

<table>
<thead>
<tr>
<th>No.</th>
<th>Course Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO 1</td>
<td>To study the Architecture of 8085 microprocessors</td>
</tr>
<tr>
<td>CO 2</td>
<td>To learn the architecture 8086 Microprocessor and its interfacing to memories</td>
</tr>
<tr>
<td>CO 3</td>
<td>To learn the instruction set of 8086 Microprocessor and assembly language programming of 8086 Microprocessor.</td>
</tr>
<tr>
<td>CO 4</td>
<td>To learn interfacing of interrupts, basic I/O and DMA with 8086 Microprocessor</td>
</tr>
</tbody>
</table>

Unit I
Evolution of Microprocessor, Introduction to 8085 - 8085 architecture - Pin Details - Addressing Modes - Instruction Set and Assembler Directives, Instruction Timing Diagram.

UNIT-II
8086 CPU ARCHITECTURE: 8086 Block diagram; description of data registers, address registers; pointer and index registers, PSW, Queue, BIU and EU. 8086 Pin diagram descriptions. Generating 8086 CLK and reset signals using 8284. WAIT state generation. Microprocessor BUS types and buffering techniques, 8086 minimum mode and maximum mode CPU module. MAIN MEMORY SYSTEM DESIGN: Memory devices, 8086 CPU Read/Write timing diagrams in minimum mode and maximum mode. Address decoding techniques. Interfacing SRAMS; ROMS/PROMS. Interfacing and refreshing DRAMS.

UNIT-III
8086 INSTRUCTION SET: Instruction formats, addressing modes, Data transfer instructions, string instructions, logical instructions, arithmetic instructions, transfer of control instructions; process control instructions; Assembler directives.

8086 PROGRAMMING TECHNIQUES: Writing assembly Language programs for logical processing, arithmetic processing, timing delays; loops, data conversions.

UNIT-IV
BASIC I/O INTERFACE: Parallel and Serial I/O Port design and address decoding. Memory mapped I/O Vs Isolated I/O Intel’s 8255 and 8251- description and interfacing with 8086. ADCs and DACs, - types, operation and interfacing with 8086. Interfacing Keyboards, alphanumeric displays, multiplexed displays, and stepper motor, optical encoder with 8086.

INTERRUPTS AND DMA: 8086 Interrupt mechanism; interrupt types and interrupt vector table. Applications of interrupts, Intel’s 8259. DMA operation. Intel’s 8237.

Text Books:
- D.V. Hall, Microprocessors and Interfacing, McGraw Hill 2nd ed.

Reference Books:
UNIT I

Introduction: Introduction to OS. Operating system functions, Different types of O.S.: batch process, multi-programmed, time-sharing, real-time, distributed, parallel.

System Structure: Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

UNIT II

CPU scheduling: scheduling criteria, preemptive & non-preemptive scheduling, scheduling algorithms, algorithm evaluation, multi-processor scheduling.

Threads: overview, benefits of threads, user and kernel threads.

Process Management: Concept of processes, process states, process control, co-operating processes, inter-process communication.

Process Synchronization: background, critical section problem, critical region, synchronization hardware, Classical problems of synchronization, semaphores.

UNIT III

Deadlocks: Concept of deadlock, deadlock characterization, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

Memory Management: background, logical vs. physical address space, contiguous memory allocation, paging, segmentation, segmentation with paging. Concept of fragmentation.

Virtual Memory: background, demand paging, concept of page replacement, page replacement algorithms, allocation of frames, thrashing.

UNIT IV

File Systems: file concept, file organization and access methods, allocation methods, directory structure, free-space management

I/O Management: I/O hardware, polling, interrupts, DMA, kernel I/O subsystem (scheduling, buffering, caching, spooling and device reservation)

Disk Management: disk structure, disk scheduling (FCFS, SSTF, SCAN,C-SCAN) , disk reliability, disk Performance parameters

Protection & Security:
Goals of protection and security, security attacks, authentication, program threats, system threats, threat monitoring.

Case studies: UNIX file system, Windows file system

Text Books:

Reference books:
4. Taub & Schilling, Principles of Communication Systems, TMH.

<table>
<thead>
<tr>
<th>CSE-212 N</th>
<th>Object Oriented Programming Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Tutorial</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Purpose: To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System

Course Outcomes (CO)

| CO 1 | To introduce the basic concepts of object oriented programming language and the its representation |
| CO 2 | To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation. |
| CO 3 | To introduce polymorphism, interface design and overloading of operator. |
| CO 4 | To handle backup system using file, general purpose template and handling of raised exception during programming |

Q1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

Q2. A point on the two dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates. Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:

Enter coordinates for P1: 3 4
Enter coordinates for P2: 5 7
Coordinates of P1 + P2 are : 8, 11

Q3. Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be ‘Y’ or ‘N’. Some sample interaction with the program might look like this.

Enter first number, operator, and second number: 10/ 3
Answer = 3.333333
Do another (Y/ N)? Y
Enter first number, operator, second number 12 + 100
Answer = 112
Do another (Y/ N) ? N

Q4. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

Enter your area code, exchange, and number: 415 555 1212
My number is (212) 767-8900
Your number is (415) 555-1212

Q5. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results maybe a DM object or DB objects, depending on the units in which the results are
required. The display should be in the format of feet and inches or metres and centimetres depending on the object on display.

Q6. Create a class rational which represents a numerical value by two double values- NUMERATOR & DENOMINATOR. Include the following public member Functions:
   • constructor with no arguments (default).
   • constructor with two arguments.
   • void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
   • Overload + operator to add two rational number.
   • Overload >> operator to enable input through cin.
   • Overload << operator to enable output through cout.
Write a main ( ) to test all the functions in the class.

Q7. Consider the following class definition
   class father {
   protected : int age;
   public;
   father (int x) {age = x;}
   virtual void iam ( )
   { cout << “I AM THE FATHER, my age is : ”<< age<< endl;}
   };
Derive the two classes son and daughter from the above class and for each, define iam ( ) to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main ( ) that creates objects of the three classes and then calls iam ( ) for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam ( ) through the pointer to demonstrate polymorphism in action.

Q8. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

Q9. A hospital wants to create a database regarding its indoor patients. The information to store include
   a) Name of the patient
   b) Date of admission
   c) Disease
   d) Date of discharge
Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

Q10. Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance variable, named department, of type string. Supply a method to String that prints the manager’s name, department and salary. Make a class Executive inherits from Manager. Supply a method to String that prints the string “Executive” followed by the information stored in the Manager superclass object. Supply a test program that tests these classes and methods.

Q11. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC key should cause the program to print out the total cars and total cash and then exit.

Q12. Write a function called reversit ( ) that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to reversit ( ) as an argument. Write a program to exercise reversit ( ). The program should get a string from the user, call reversit ( ), and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon’s famous phrase, “Able was I ere I saw Elba”.

Q13. Create some objects of the string class, and put them in a Deque-some at the head of the Deque and some at the tail. Display the contents of the Deque using the forEach ( ) function and a user written display function. Then search the Deque for a particular string, using the first That ( ) function and display any strings that match. Finally remove all the items from the Deque using the getLeft ( ) function and display each item. Notice the order in which the items are displayed: Using getLeft ( ), those inserted on the left (head) of the Deque are
removed in “last in first out” order while those put on the right side are removed in “first in first out” order. The opposite would be true if `getRight()` were used.

**Q14.** Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class account that stores customer name, account number and type of account. From this derive the classes `cur_acct` and `sav_acct` to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

a) Accept deposit from a customer and update the balance.
b) Display the balance.
c) Compute and deposit interest.
d) Permit withdrawal and update the balance.
e) Check for the minimum balance, impose penalty, necessary and update the balance.
f) Do not use any constructors. Use member functions to initialize the class members.

**Q15.** Create a base class called `shape`. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called `triangle` and `rectangle` from the base `shape`. Add to the base class, a member function `get_data()` to initialize base class data members and another member function `display_area()` to compute and display the area of figures. Make `display_area()` as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle = x * y
Area of triangle = \( \frac{1}{2} \) * x * y
Write an Assembly Language Program to
1. Add / Sub two 16 bit numbers.
2. Find sum of series of numbers.
3. Multiply two 16 bit unsigned/ signed numbers.
4. Divide two unsigned/ signed numbers (32/16, 16/8, 16/16, 8/8)
5. Add / Sub / multiply / Divide two BCD numbers.
6. Find smallest/ largest number from array of n numbers.
7. Arrange numbers in array in ascending/ descending order.
8. Perform block transfer data using string instructions / without using string instructions.
9. Compare two strings using string instructions / without using string instructions.
10. Display string in reverse order, string length, Concatenation of two strings.
11. Convert Hex to Decimal, Decimal to Hex.
12. To find 1’s and 2’s complement of a number.
CSE-216 N  Internet Lab

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Minor Test</th>
<th>Practical</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3 Hour</td>
</tr>
</tbody>
</table>

Purpose: Learn the internet and design different web pages using HTML and installation of different MODEMS.

Course Outcomes

| CO 1 | Understanding different PC software and their applications |
| CO 2 | To be able to learn HTML. |
| CO 3 | To be able to write Web pages using HTML. |
| CO 4 | To be able to install modems and understand the e-mail systems. |

1. To prepare the Your Bio Data using MS Word
2. To prepare the list of marks obtained by students in different subjects and show with the help of chart/graph the average, min and max marks in each subject.
3. Prepare a presentation explaining the facilities/infrastructure available in your college/institute.

HTML Lists:
1. Create a new document that takes the format of a business letter. Combine `<P>` and `<BR>` tags to properly separate the different parts of the documents. Such as the address, greeting, content and signature. What works best for each?
2. Create a document that uses multiple `<BR>` and `<P>` tags, and put returns between `<PRE>` tags to add blank lines to your document see if your browser senders them differently.
3. Create a document using the `<PRE>`tags to work as an invoice or bill of sale, complete with aligned dollar values and a total. Remember not to use the Tab key, and avoid using emphasis tags like `<B>` or `<EM>` within your list.
4. Create a seven-item ordered list using Roman numerals. After the fifth item, increase the next list value by 5.
5. Beginning with an ordered list, create a list that nests both an unordered list and a definition list.
6. Use the ALIGN attribute of an `<IMG>` tags to align another image to the top of the first image. play with this feature, aligning images to TOP, MIDDLE and BOTTOM.
7. Create a ‘table of contents’ style page (using regular and section links) that loads a different document for each chapter or section of the document.

Internet:
1. Instilling internet & external modems, NIC and assign IP address.
2. Study of E-mail system.
3. Create your own mail-id in yahoo and indiatimes.com.
4. Add names (mail-id’s) in your address book, compose and search an element.
UNIT I


(a) Forest Resources: Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forests and tribal people.

(b) Water Resources- Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.

(c) Mineral Resources- Use and exploitation, environmental effects of extracting and using mineral resources, case studies.

(d) Food Resources- World Food Problems, changes caused by agriculture and overgazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.

(e) Energy Resources- Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources. Case studies.

(f) Land Resources- Land as a resource, land degradation, man induced landslides, soil erosion and desertification.

Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyle.

UNIT II

Ecosystem-Concept of an ecosystem. Structure and function of an ecosystem. Producers, consumers and decomposers. Energy flow in the ecosystem. Ecological Succession. Food Chains, food webs and ecological pyramids. Introduction, types, characteristic features, structure and function of the following ecosystem-

a. Forest Ecosystem

b. Grassland Ecosystem
c. Desert Ecosystem
d. Aquatic Ecosystems(ponds, streams, lakes, rivers, oceans, estuaries

Field Work. Visit to a local area to document Environment assets-river/forest/grassland/hill/mountain. Visit to a local polluted site- Urban /Rural Industrial/Agricultural. Study of common plants, insects and birds. Study of simple ecosystems-pond, river, hill, slopes etc. (Field work equal to 5 lecture hours).

UNIT III


Environmental Pollution Definition. Cause, effects and control measures of- (a) Air Pollution (b) Water Pollution (c) Soil Pollution (d) Marine Pollution (e) Noise Pollution (f) Thermal Pollution (g) Nuclear Hazards Solid waste management- cause, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: floods, earthquake, cyclone and landslides.

UNIT IV

**Text Books**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course No.</th>
<th>Subject</th>
<th>L:T:P</th>
<th>Hours/Week</th>
<th>Examination Schedule (Marks)</th>
<th>Duration of Exam (Hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Major Test</td>
<td>Minor Test</td>
</tr>
<tr>
<td>1</td>
<td>CSE 301N</td>
<td>Automata Theory</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>CSE 303 N</td>
<td>Computer Networks</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>CSE 305N</td>
<td>Design and Analysis of algorithms</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>CSE 307N</td>
<td>Computer organisation and Architecture</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>CSE 309N</td>
<td>Simulation &amp; Modelling</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>CSE 311N</td>
<td>Computer Networks Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>CSE 313N</td>
<td>Design and Analysis of algorithms Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>CSE 315 N</td>
<td>Simulation Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>CSE 317N</td>
<td>Seminar/Industrial Training*</td>
<td>0:0:2</td>
<td>2</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>10</td>
<td>CSE 319N</td>
<td>Technical Communication and Soft Skills Lab</td>
<td>0:0:2</td>
<td>2</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td>375</td>
</tr>
</tbody>
</table>
### Lecture Outline

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3 Hrs.</td>
</tr>
</tbody>
</table>

**Purpose**
To understand the challenges for Theoretical Computer Science and its contribution to other sciences.

<table>
<thead>
<tr>
<th>Course Outcomes (CO)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CO1</strong> Students are able to explain and manipulate the different fundamental concepts in automata theory and formal languages.</td>
</tr>
<tr>
<td><strong>CO2</strong> Simplify automata and context-free grammars, Prove properties of languages, grammars and automata with rigorously formal mathematical methods, minimization.</td>
</tr>
<tr>
<td><strong>CO3</strong> Differentiate and manipulate formal descriptions of push down automata, its applications and transducer machines.</td>
</tr>
<tr>
<td><strong>CO4</strong> To understand basic properties of Turing machines and computing with Turing machine, the concepts of tractability and decidability.</td>
</tr>
</tbody>
</table>

---

**Unit - 1**

**Introduction to Automata**: Study and Central Concepts of Automata Theory, Applications of Finite Automata, An Introduction of Deterministic Finite Automata (DFA) and Non-Deterministic Finite Automata (NFA), Finite Automata with Epsilon (€) Transitions.

**Regular Expression and Languages**: Regular Expressions (RE), Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws of Regular Expressions, Closure Properties of Regular Languages, RE to NFA, DFA Conversion and DFA to RE, Equivalence and Minimization of NFA and DFA automata.

**Unit-2**


**Pumping Lemma**: Introduction to Pumping Lemma, pumping lemma for context free languages, Applications of Pumping Lemma, Minimization of Finite Automata, and Recursive Language.

**Unit-3**

**Mealey and Moore Machines**: Definitions, Representation, Equivalence of Moore and Mealey Machines and its Designing.

**Push Down Automata**: Introduction of Push Down Automata (PDA), Language of PDA, Equivalence of PDA's and CFG's, Deterministic Push Down Automata, Designing of PDA, Applications of PDA. Parikh Theorem and Parikh Mapping, Kleene's Theorem.

**Unit-4**

**Introduction to Turing Machine**: The Turing Machine, Programming Techniques for Turing Machine, Extensions of Turing Machine, Restricted Turing Machines, Universal Turing Machines and Designing of Turing Machines, Time and Tape Complexity Measures of Turing machines.

**Decidability**: Post's Correspondence Problem (PCP), Rice's Theorem, Decidability of Membership, Emptiness and Equivalence Problems of Languages.

---

**Textbooks**

**References**
Purpose
To introduce the architecture and layers of computer network, protocols used at different layers.

Course Outcomes (CO)

<table>
<thead>
<tr>
<th>CO</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>To understand the basic concept of networking, types, networking topologies and layered architecture.</td>
</tr>
<tr>
<td>CO2</td>
<td>To understand data link layer and MAC sub-layer</td>
</tr>
<tr>
<td>CO3</td>
<td>To understand the network Layer functioning</td>
</tr>
<tr>
<td>CO4</td>
<td>To understand the transport layer and application layer operation</td>
</tr>
</tbody>
</table>

Unit -1


Unit -2

Data link layer: Error Control, Types of errors, framing (character and bit stuffing), error detection & correction methods; Flow control; Protocols: Stop & wait ARQ, Go-Back- N ARQ, sliding window protocols, Selective repeat ARQ, HDLC

Medium access sub layer: Point to point protocol, FDDI, token bus, token ring; Reservation, polling, Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA, CSMA/CD, FDMA, TDMA, CDMA, LLC, Traditional Ethernet, fast Ethernet, Network devices-repeaters, hubs, switches, Bridges, Router, Gateway

Unit -3

Network layer: Addressing: Internet address, subnetting; Routing techniques, static vs. dynamic routing, routing table, DHCP, IEEE standards 802.x, Routing algorithms: shortest path algorithm, flooding, distance vector routing, link state routing; Protocols: ARP, RARP, IP, ICMP, IGMP, IPV6; Unicast and multicast routing protocols.

Unit -4

Transport layer: Process to process delivery; UDP; TCP, RPC, Congestion control algorithm: Leaky bucket algorithm, Token bucket algorithm, choke packets; Quality of service: techniques to improve QoS.

Application layer: DNS; SMTP, SNMP, FTP, HTTP & WWW; Firewalls, Bluetooth, Email, S/MIME, IMAP


TEXT BOOK

REFERENCES
Lecture | Tutorial | Practical | Major Test | Minor Test | Total | Time
---|---|---|---|---|---|---
3 | 1 | - | 75 | 25 | 100 | 3 Hrs.

**Purpose**
To introduce advanced data structures & algorithms concepts involving their implementation for solving complex applications.

**Course Outcomes (CO)**

| CO1 | Learn the basic concepts of data structures and their analysis. |
| CO2 | Study the concept of dynamic programming and various advanced data structures. |
| CO3 | Learn various graph algorithms and concepts of computational complexities. |
| CO4 | Study various Flow and Sorting Networks |

**Unit 1**

**Introduction**

**Review:** Elementary Data Structures, Algorithms & its complexity(Time & Space), Analysing Algorithms, Asymptotic Notations, Priority Queue, Quick Sort and merge sort.

**Recurrence relation:** Methods for solving recurrence(Substitution, Recursion tree, Master theorem), Strassen multiplication.

**Advanced data Structures:** Binomial heaps, Fibonacci heaps, Splay Trees, Red-Black Trees.

**Unit 2**

**Advanced Design and analysis Techniques**

**Dynamic programming:** Elements, Matrix-chain multiplication, longest common subsequence,

**Greedy algorithms:** Elements, Activity- Selection problem, Huffman codes, Task scheduling problem, Travelling Salesman Problem.

**Backtracking algorithms:** Graph coloring, N-Queen problem, Hamiltonian path and circuit.

**Unit 3**

**Graph Algorithms**

Review of graph algorithms:Traversal Methods(Depth first & Breadth first search),Topological sort, Strongly connected components, Minimum spanning trees- Kruskal’s and Prim’s Algorithm, Single source shortest paths, Relaxation, Dijkstra’s Algorithm, Bellman- Ford algorithm, Single source shortest paths for directed acyclic graphs, Floyd-Warshall algorithm.

**Unit 4**

**Computational Complexity:** Basic Concepts, Polynomial vs Non-Polynomial Complexity, NP-hard & NP-complete classes. Flow and Sorting Networks, Flow networks, Ford- Fulkerson method, Maximum bipartite matching, Sorting Networks, Comparison network, Zero- one principle, Bitonic sorting network, merging network

**Text Books:**
1. Corman, Leiserson and Rivest : Introduction to Algorithms, 2/e, PHI

**Reference Books:**
CSE-307N Computer Organization and Architecture

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3 Hrs.</td>
</tr>
</tbody>
</table>

**Purpose**

Student will be able to understand the basic concepts of computer architecture and organization, and understand the key skills of constructing cost-effective computer systems.

**Course Outcomes (CO)**

<table>
<thead>
<tr>
<th>CO</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>Be familiar with the functional units of the processor such as the register file and arithmetic-logical unit, and with the basics of systems topics</td>
</tr>
<tr>
<td>CO2</td>
<td>Be familiar with the design trade-offs in designing and constructing a computer processor.</td>
</tr>
<tr>
<td>CO3</td>
<td>Be familiar with the CPU design including the RISC/CISC architectures.</td>
</tr>
<tr>
<td>CO4</td>
<td>Be familiar with the basic knowledge of I/O devices and interfacing of I/O devices with computer.</td>
</tr>
</tbody>
</table>

**Unit- I**

**Data representation and Computer arithmetic:** Introduction to Computer Systems, Organization and architecture, evolution and computer generations; Fixed point representation of numbers, digital arithmetic algorithms for Addition, Subtraction, Multiplication using Booth’s algorithm and Division using restoring and non restoring algorithms. Floating point representation with IEEE standards and its arithmetic operations.

**Unit-II**

**Basic Computer organization and Design:** Instruction codes, stored program organization, computer registers and common bus system, computer instructions, timing and control, instruction cycle: Fetch and Decode, Register reference instructions; Memory reference instructions. Input, output and Interrupt: configuration, instructions, Program interrupt, Interrupt cycle, Micro programmed Control organization, address sequencing, micro instruction format and microprogram sequencer.

**Unit-III**

**Central Processing Unit:** General register organization, stack organization, instruction formats, addressing modes, Data transfer and manipulation, Program control. CISC and RISC: features and comparison. Pipeline and vector Processing, Parallel Processing, Pipelining, Instruction Pipeline, Basics of vector processing and Array Processors.

**Unit-IV**


**TEXT BOOK:**


**REFERENCES:**

Unit-1

**Modeling:** System Concepts, system boundaries and environment, continuous and discrete systems, system modeling, types of Models, Model validation, Principles & Nature of Computer modeling.

**Simulation:** Introduction, Basic nature of simulation, when to simulate, Advantages, disadvantages and limitations of simulation, Concepts of simulation of continuous and discrete system with the help of example.

Unit-2

**Continuous System Simulation:** Analog vs. digital simulation, continuous simulation vs. numerical integration, simulation of a chemical reactor, simulation of a water reservoir system.

**Discrete system simulation:** Fixed time-step vs. event-to-event model, Monte-Carlo computation vs. stochastic simulation, generation of random numbers, and generation of non-uniformly distributed random numbers.

Unit -3

**Simulators for the Live systems:** Simulation of queueing Systems: basic concepts of queueing theory, simulation of single server, two server and more general queueing system.

**Simulation of PERT network:** Network model of a project, analysis of an activity network, critical path computation, uncertainties in activity durations, simulation of an activity network.

Unit-4

**Simulation of inventory control systems:** Elements of inventory theory, inventory models, generation of Poisson and Erlang variates, simulator for complex inventory systems.

**Simulation of hypothetical computers.**

**Design and Evaluation of Simulation Experiments:** Variance reduction techniques. Experiment layout and Validation.

**Case Study:** SciLab, Octave.

Text Books:
2. Narsingh Deo: System Simulation with Digital Computer, PHI New Delhi, 1993

Reference Books:

<table>
<thead>
<tr>
<th>CSE-311N</th>
<th>Computer Networks Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Tutorial</td>
</tr>
<tr>
<td></td>
<td>Practical</td>
</tr>
<tr>
<td></td>
<td>--</td>
</tr>
<tr>
<td>Purpose</td>
<td>To explore networking concepts using Java programming &amp; networking tools.</td>
</tr>
</tbody>
</table>

**Course Outcomes (CO)**

1. Do Problem Solving using algorithms.
2. Design and test simple programs to implement networking concepts using Java.
4. Design simple data transmission using networking concepts and implement.

**COMPUTER NETWORKS (Lab)**

1. Create a socket for HTTP for web page upload and download.
2. Write a code simulating ARP /RARP protocols.
4. Performance comparison of MAC protocols
5. Performance comparison of routing protocols.
6. Write a program:
   - To implement echo server and client in java using TCP sockets.
   - To implement date server and client in java using TCP sockets.
   - To implement a chat server and client in java using TCP sockets.
7. Write a program:
   - To implement echo server and client in java using UDP sockets
   - To implement a chat server and client in java using UDP sockets.
   - To implement a DNS server and client in java using UDP sockets.
8. To flood the server from a spoofed source address leading to a DoS attack.
9. To sniff and parse packets that pass through using raw sockets.
10. To implement simple calculator and invoke arithmetic operations from a remote client.
11. To implement bubble sort and sort data using a remote client.
12. To simulate a sliding window protocol that uses Go Back N ARQ.
The student will learn the algorithm analysis techniques, become familiar with the different algorithm design techniques and understand the limitations of Algorithm power.

Course Outcomes (CO)

CO1 The student should be able to design algorithms for various computing problems.

CO2 The student should be able to analyse the time and space complexity of algorithms.

CO3 The student should be able to critically analyse the different algorithm design techniques for a given problem.

CO4 The student should be able to modify existing algorithms to improve efficiency.

List of Practical

1. Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of \( n \), the number of elements in the list to be sorted and plot a graph of the time taken versus \( n \). The elements can be read from a file or can be generated using the random number generator.

2. Using Open, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of \( n \), the number of elements in the list to be sorted and plot a graph of the time taken versus \( n \). The elements can be read from a file or can be generated using the random number generator.

3. a. Obtain the Topological ordering of vertices in a given digraph.
   b. Compute the transitive closure of a given directed graph using Warshall's algorithm.

4. Implement 0/1 Knapsack problem using Dynamic Programming.

5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

7. a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
   b. Check whether a given graph is connected or not using DFS method.

8. Find a subset of a given set \( S = \{s_1, s_2, \ldots, s_n\} \) of \( n \) positive integers whose sum is equal to a given positive integer \( d \). For example, if \( S = \{1, 2, 5, 6, 8\} \) and \( d = 9 \) there are two solutions \( \{1, 2, 6\} \) and \( \{1, 8\} \). A suitable message is to be displayed if the given problem instance doesn't have a solution.

9. Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.


11. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm. Parallelize this algorithm, implement it using Open and determine the speed-up achieved.

12. Implement N Queen's problem using Back Tracking.

13. Implement Graph Coloring.

14. Find Hamiltonian Path using Back Tracking.

15. Implement longest common subsequence.

CSE 315N  Simulation lab

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Minor Test</th>
<th>Practical</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>40</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 Hour</td>
</tr>
</tbody>
</table>

Purpose: To introduce the principles and paradigms of Computer Simulation for solving a wide variety of problems. In addition, how to use simulator to simulate the live systems.

Course Outcomes (CO)

<table>
<thead>
<tr>
<th>CO1</th>
<th>Learn the simulation of continuous and discrete systems with the help of different examples.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>Learn the concept of generation of uniformly and non-uniformly distributed random numbers.</td>
</tr>
<tr>
<td>CO3</td>
<td>Learn the simulation of queuing system.</td>
</tr>
<tr>
<td>CO4</td>
<td>Learn the concept of simulation CPM and PERT.</td>
</tr>
<tr>
<td>CO5</td>
<td>Learn the concept of simulation of inventory control system.</td>
</tr>
</tbody>
</table>

LIST OF EXPERIMENTS

1: Write a program to print the detailed marks certificate (D.M.C) of a student by using different binary operators.
2: Write a program to Draw graph of sine wave with respect to the time.
3: Write a program to solve following differential equation
   \[ \frac{dy}{dt} = -\exp(-t) \times y^2 \] by using any simulation technique.
4: Write a program to solve following differential equation by using 4th order Runge-Kutta method
   \[ \frac{dy}{dx} = -2x-y \] , with initial condition \( y = -2 \) when \( x = 0 \).
5: Write a program to simulate Pure-Pursuit problem of continuous system simulation.
6: Write a program to select a policy among different given policies with minimum total cost of an inventory system.
7: Write a program to generate and print a sequence of 30 pseudo random numbers between 150 to 250 by using any simulation technique.
8: Write a program to determine the approximate value of \( \sqrt{2} \) using 1000 random numbers.
9: Write a program to generate a sample of pseudo random values by using rejection method from a given non-uniform distribution, when the probability function of the distribution is non-zero over finite interval (a, b).
10: Write a program to simulate single server queuing system with Poisson arrival pattern and FCFS queue discipline.
11: Write a program to find minimum time of completing the project by PERT.
12: Write a program to simulate an inventory system with the objective to determine the reorder combination (P,Q) which yields the highest service level for a given value of average stock.
### Technical Communication and Soft Skills Lab

<table>
<thead>
<tr>
<th>CSE-319N</th>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>3 Hours</td>
</tr>
</tbody>
</table>

**Purpose**

To enhance the students’ oral communication skills in English

#### Course Outcomes (CO)

<table>
<thead>
<tr>
<th>CO1</th>
<th>Develop oral communicative competence in English</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>Improve fluency in English and thereby respond confidently due to reduced communication apprehension</td>
</tr>
<tr>
<td>CO3</td>
<td>Identify and explain the biological and physiological characteristic of proper voice and diction production</td>
</tr>
<tr>
<td>CO4</td>
<td>Develop correct and better pronunciation through stress on word accent, intonation, and weak forms</td>
</tr>
<tr>
<td>CO5</td>
<td>Participate in Group Discussions effectively</td>
</tr>
<tr>
<td>CO6</td>
<td>Make effective oral presentations in English</td>
</tr>
</tbody>
</table>

#### LIST OF TOPICS FOR LAB ACTIVITIES

The following topics are prescribed to conduct the activities in the lab:

1. Articulation of Consonant sounds
2. Articulation of Vowel sounds
3. Pronunciation
4. Word Accent
5. Weak Forms
6. Intonation
7. Conversation in different formal situations
8. Group Discussion
9. Oral presentation
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course No.</th>
<th>Subject</th>
<th>L:T:P</th>
<th>Hours/Week</th>
<th>Examination Schedule (Marks)</th>
<th>Duration of Exam (Hrs)</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Practical</th>
<th>Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CSE 302N</td>
<td>Compiler Design</td>
<td>3:1:0</td>
<td>4</td>
<td></td>
<td></td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>CSE 304N</td>
<td>Essential of Information Technology</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CSE 306N</td>
<td>Mobile Computing</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CSE 308N</td>
<td>Web Technology</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CSE 310N</td>
<td>Software Engineering</td>
<td>3:1:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>HS 303N</td>
<td>Business</td>
<td>4:0:0</td>
<td>4</td>
<td>75</td>
<td>25</td>
<td>0</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course</td>
<td>Description</td>
<td>Lecture</td>
<td>Tutorial</td>
<td>Practical</td>
<td>Major Test</td>
<td>Minor Test</td>
<td>Total</td>
<td>Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
<td>----------</td>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
<td>-------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSE 312N</td>
<td>Web Technology Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSE 314N</td>
<td>Essential of Information Technology Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSE 316N</td>
<td>Software Engineering Lab</td>
<td>0:0:3</td>
<td>3</td>
<td>0</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>33</td>
<td>450</td>
<td>270</td>
<td>180</td>
<td>900</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Purpose: At the end of the course, the student will be able to design and implement a compiler.

Course Outcomes (CO):
- CO1: To understand, design and implement a lexical analyzer.
- CO2: To understand, design and implement a parser.
- CO3: To understand, design code generation schemes.
UNIT I

Introduction to Compiling
Analysis of the source program, Phases of a compiler, Cousins of the Compiler, Grouping of Phases, Compiler construction tools.
Lexical Analysis –Regular Expression, Introduction to Finite Automata and Regular Expression, Conversion of Regular Expression to NFA, Role of Lexical Analyzer, Input Buffering, Specification of Tokens.

UNIT II

Syntax Analysis
Role of the Parser, Writing Grammars, Symbol Table, Context-Free Grammars, Top Down Parsing with or without Backtracking, Recursive Descent Parsing, Non-Recursive Descent Parsing, SLR Parser, Canonical LR Parser, LALR Parser.

UNIT III

Intermediate Code Generation and Code
Intermediate languages, Declarations, Assignment Statements, Boolean Expressions, Case Statements, DAG representation of Basic Blocks, A simple Code generator from DAG, Issues in the design of code generator , The target machine , Runtime Storage management, Error Handling- Type checking,

UNIT IV

Code Optimization and Run Time Environments

TEXT BOOK

REFERENCES
**Focus Area 1: Object Oriented Programming using Java**

**Unit I:**

**Problem Solving Techniques:** Introduction to problem solving, Computational problem and its classification - Logic and its types, Introduction to algorithms and flowchart, Searching algorithms: linear search, binary search and sorting algorithms: insertion, quick, merge and selection sort, Introduction and classification to Data Structures, Basic Data Structures: array, stack, and queue.

**Unit II:**

**Programming Basics:** Identifiers, variables, data types, operators, control structures, type conversion, casting, arrays, strings

**Object Oriented Concepts fundamentals:** class & object, instance variables & methods, access specifiers, reference variables, parameter passing techniques, constructors, this reference, static, and command line arguments

**Introduction to UML:** Use case diagrams – Class diagrams

**Unit III:**

**Relationships:** aggregation, association, Inheritance, types of inheritance, Static Polymorphism: method overloading, constructor overloading, Dynamic polymorphism: method overriding, abstract, interface, introduction to packages Industry Coding Standards and Best Practices, code tuning & optimization, clean code & refactoring

---

**Focus Area 2: Relational Database Management System**

**Unit IV:**

RDBMS- data processing, the database technology, data models, ER modelling concept, notations, converting ER diagram into relational schema, Logical database design, normalization (1NF, 2NF and 3NF)

SQL: DDL statements, DML statements, DCL statements, Joins, Sub queries, Views, Database design Issues, SQL fine-tuning

**Books on Java**
2. Programming with *Java 3e A Primer* by E Balagurusamy
3. Introduction to Java Programming by K. Somasundaram , Jaico Publishing House; 1 edition

**Books on RDBMS, Oracle, MYSQL**
5. Schaum's Outline of Fundamentals of Relational Databases by Ramon Mata-Toledo, Published November 15th 2000 by McGraw-Hill
Purpose
To impart knowledge of mobile and wireless computing systems and techniques.

Course Outcomes (CO)

| CO1 | Describe the concepts of mobile computing and cellular networks. |
| CO2 | Learn the basic concepts of wireless networks. |
| CO3 | Study of various issues of mobile computing and basics of cloud computing. |
| CO4 | Description and applications of Ad hoc networks. |

UNIT - I
Introduction, issues in mobile computing, overview of wireless telephony: cellular concept, Mobile computing Architecture, Design considerations for mobile computing, Mobile Computing through Internet, Making existing applications mobile enabled. GSM: air-interface, channel structure, location management: HLR-VLR, hierarchical, handoffs, channel allocation in Cellular systems, WCDMA, GPRS 3G, 4G.

UNIT - II

UNIT - III
Data management issues, data replication for mobile computers, adaptive clustering for mobile wireless networks, File system, Disconnected operations Mobile Agents computing, security and fault tolerance, transaction processing in mobile computing environment, Cloud Architecture model, Types of Clouds: Public Private & Hybrid Clouds, Resource management and scheduling, Clustering, Data Processing in Cloud: Introduction to Map Reduce for Simplified data processing on Large clusters.

UNIT - IV
Ad hoc networks, localization, MAC issues, Routing protocols, global state routing (GSR), Destination sequenced distance vector routing (DSDV), Dynamic source routing (DSR), Ad Hoc on demand distance vector routing (AODV), Temporary ordered routing algorithm (TORA), QoS in Ad Hoc Networks, applications.

Text Books:
2. J. Schiller, Mobile Communications, Addison Wesley

Reference Books
1. A. Mehrotra, GSM System Engineering.
CSE-308N
Web Engineering

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Major Test</th>
<th>Minor Test</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

Purpose
To gain a broad understanding of the discipline of Web engineering and its application to the development and management of Web Applications.

Course Outcomes

<table>
<thead>
<tr>
<th>Course Outcomes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>Learn the basic concepts of information and web architecture.</td>
</tr>
<tr>
<td>CO2</td>
<td>Learn about the skills that will enable to design and build high level web enabled applications.</td>
</tr>
<tr>
<td>CO3</td>
<td>Understand the applicability of Java Script as per current software industry standards.</td>
</tr>
<tr>
<td>CO4</td>
<td>Acquaint the latest programming language for the implementation of object based and procedure based applications using Python.</td>
</tr>
</tbody>
</table>

Unit-1
Information Architecture: The role of Information Architect, Collaboration and communication, Organizing information, organizational challenges, Organizing web sites and Intranets, Creating cohesive organization systems, designing navigation systems, types of navigation systems, Integrated navigation elements, designing elegant navigation systems, Searching systems, Searching your web site, designing the search interface, Indexing the right stuff, To search or not to search grouping content, conceptual design, High level Architecture Blueprint. Architectural Page Mockups, Design Sketches.

Unit-2
Introduction to XHTML and HTML5: Origins and Evolution of HTML and XHTML, Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML5, Syntactic Differences between HTML and XHTML.
Cascading Style Sheets: Introduction, Levels of Style Sheets, Style Specification Formats, Selector Forms, Property Value Forms, Font Properties, List Properties, Color, Alignment of Text, Box Model, Background Images, Conflict Resolution.

Unit-3

Unit-4
Python: Introduction to Python, Data Types and Expressions, Control Statements, Strings and Text Files, Lists and Dictionaries, Design with Functions, Design with Classes.

Text Books

Reference Book
Purpose: To gain a broad understanding of the discipline of software engineering and its application to the development and management of software process.

Course Outcomes (CO):

CO1: To understand the basic concepts of Software Engineering.
CO2: To learn about the skills that will enable to construct high quality software.
CO3: To understand the software process models.
CO4: To understand the fundamental concept of requirements engineering and Analysis Modelling.
CO5: To understand the different design techniques and their implementation.
CO6: To learn about software testing and maintenance measures.

Unit-I


Unit-II

Software Quality: Software Quality, Concept of Software Quality Assurance (SQA), SEI-CMM Model.
Introduction to Software Risk Management and Software Configuration Management

Unit-III


Unit-IV

Software Construction: Software construction fundamentals, minimizing complexity, Top-Down and Bottom –Up programming, structured programming, Compliance with Design and Coding Standards.
Maintenance: key issues, Types of software Maintenance, Cost of Maintenance, Software Re-Engineering.

Text Books:

Reference Books:
1. Pankaj Jalote, Software Engineering, Wiley India.

<table>
<thead>
<tr>
<th>HS-303N</th>
<th>Business Intelligence &amp; Entrepreneurship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Tutorial</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
</tr>
</tbody>
</table>

Course Outcomes
CO1 Students will be able understand who the entrepreneurs are and what competences needed to become an Entrepreneur
CO2 Students will be able understand insights into the management, opportunity search, identification of a Product; market feasibility studies; project finalization etc. required for small business enterprises.
CO3 Students can be able to write a report and do oral presentation on the topics such as product identification, business idea, export marketing etc.
CO4 Students be able to know the different financial and other assistance available for the establishing small industrial units.

Unit -I
Entrepreneurship: Concept and Definitions; Entrepreneurship and Economic Development; Classification and Types of Entrepreneurs; Entrepreneurial Competencies; Factor Affecting Entrepreneurial Growth – Economic, Non-Economic Factors; EDP Programmes; Entrepreneurial Training; Traits/Qualities of an Entrepreneurs; Entrepreneur; Manager Vs. Entrepreneur.

Unit -II
Opportunity / Identification and Product Selection: Entrepreneurial Opportunity Search and Identification; Criteria to Select a Product; Conducting Feasibility Studies; Project Finalization; Sources of Information.

Unit -III
Small Enterprises and Enterprise Launching Formalities : Definition of Small Scale; Rationale; Objective; Scope; Role of SSI in Economic Development of India; SSI; Registration; NOC from Pollution Board; Machinery and Equipment Selection; Project Report Preparation; Specimen of Project Report; Project Planning and Scheduling using Networking Techniques of PERT / CPM; Methods of Project Appraisal.

Unit -IV
Role of Support Institutions and Management of Small Business : Director of Industries; DIC; SIDO; SIDBI; Small Industries Development Corporation (SIDC); SISI; NSIC; NISBUD; State Financial Corporation SIC; Marketing Management; Production Management; Finance Management; Human Resource Management; Export Marketing; Case Studies-At least one in whole course.
Text Books:


<table>
<thead>
<tr>
<th>CSE-312N</th>
<th>Web Engineering Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Tutorial</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Purpose</td>
<td>To introduce the concepts of HTML5, JavaScript and Python.</td>
</tr>
</tbody>
</table>

**Course Outcomes (CO)**

- **CO1** Design webpages using HTML, JavaScript and CSS.
- **CO2** Design and test simple function/program to implement Searching and sorting techniques using Python.
- **CO3** Develop program in Java Script for pattern matching using regular expressions and errors in scripts.
- **CO4** Design client-server based web applications.

[1] Create your own page with your favorite hobbies using HTML, JavaScript and CSS.
[2] Create a frameset in HTML that is divided into three sections. The frameset should have three zones.
  a. The Topmost section of the frameset should take up about just 15% of the browser window. Name this frame title.
  b. The middle section should be 75% of the browser window. Name this frame title.
  c. The lower section should be 10% of the browser window. Name this frame menu.
[3] Create pages for each section. For the lowermost section, create a page that loads the content into the middle section. The topmost section should contain a page describing the web page itself.
[4] Create a web page, which displays the map of your country Link, each city /state on the image map, such that the respective HTML page of the city/state is displayed when the user selects an area.
[5] Add the tickertape applet to your page by customizing it for the following settings:
  a. Increase the count by one.
  b. Accordingly update the message count.
  c. Change the text color to (237,192,171)
  d. Experiment with changing the scrolling speed.
  e. Customize the message text as per your page requirement.
[7] Use Cascading Style sheets (CSS) to modify the following:
  a. Change background.
  b. Change font type, face and color.
  c. Align Text.
d. Remove underlines from hyperlinks.

[8] Write the program for using JavaScript by using for – loops (through a block of code a number of times), for/in - loops (through the properties of an object), while - loops (through a block of code while a specified condition is true), do/while - loops (through a block of code while a specified condition is true).

[9] Write a program in Java Script for the following:
   a. Copying, passing, and comparing by value
   b. Copying, passing, and comparing by reference
   c. References themselves are passed by value


[11] Write a Python function/program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is an equilateral triangle.


[13] Write program in Python using Lists and dictionaries, Control statements and Strings and text files.

---

<table>
<thead>
<tr>
<th>CSE-314N</th>
<th>Essentials of Information Technology Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Tutorial</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Purpose</td>
<td>To introduce the concepts of Object Oriented Programming using Java and RDBMS</td>
</tr>
</tbody>
</table>

Course Outcomes (CO)

| CO1       | Do Problem Solving using algorithms |
| CO2       | Design and test simple programs to implement Object Oriented concepts using Java |
| CO3       | Document artifacts using common quality standards |
| CO4       | Design simple data store using RDBMS concepts and implement |

Students should implement at least 4-5 problems from the real world related to concern engineering branch for following both focus area during Practical hours:

1. Programs using Java Language
2. RDBMS Queries using MySQL

Tools:

- Understanding basic programming constructs using Scratch Tool - Flowcharts implementation through RAPTOR tool
- Eclipse IDE for Java programming
Software Engineering Lab

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Tutorial</th>
<th>Practical</th>
<th>Minor Test</th>
<th>Practical</th>
<th>Total</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>3</td>
<td>40</td>
<td>60</td>
<td>100</td>
<td>3</td>
</tr>
</tbody>
</table>

Purpose
To gain a broad understanding of the discipline of software engineering implementation.

Course Outcomes

CO1  To understand the basic concepts of Software Engineering.
CO2  To learn about the reasons for the software crisis.
CO3  To understand the software testing techniques.
CO4  To understand the software metrics.
CO5  To understand the different design techniques and their implementation.
CO6  To learn about software testing and maintenance measures.

List of Practical’s

1. To identify the role of the software in today’s world across a few significant domains related to day to day life.
2. To identify the problem related to software crisis for a given scenario.
3. To classify the requirement into functional and non-functional requirements.
4. To implement at least four software metrics.
5. Preparation of requirement document for standard application problems in standard format. (e.g Library Management System, Railway Reservation system, Hospital management System, University Admission system)
6. To prepare Project Schedule for standard application problems in standard format.
7. To implement the functional testing techniques.
8. To implement the structural testing techniques.
### Bachelor of Technology (Computer Engineering)
#### Schemes of Studies / Examination (Semester- 8TH)

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course No.</th>
<th>Subject</th>
<th>Teaching Schedule</th>
<th>Examination Schedule</th>
<th>Duration Of Exam (Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>Departmental Elective-IV</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>**</td>
<td>Departmental Elective-V</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CSE-402</td>
<td>Neural Networks &amp; Fuzzy Logic</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CSE-404</td>
<td>Interactive Computer Graphics</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CSE-406</td>
<td>Neural Networks (Pr.)</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>CSE-408</td>
<td>Major Project</td>
<td>-</td>
<td>-</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>CSE-410</td>
<td>Seminar</td>
<td>-</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>CSE-412</td>
<td>Comprehensive Viva-Voce</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>CSE-414</td>
<td>General Fitness &amp; Professional Aptitude</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td>14</td>
<td>6</td>
<td>15</td>
</tr>
</tbody>
</table>

**Departmental Elective-IV**

1. CSE-440 Distributed Operating Systems  
2. CSE-442 Software Quality Models and Testing  
3. CSE-444 Bioinformatics  
4. CSE-446 Expert Systems  
5. CSE-448 Real Time Systems and Softwares  
6. CSE-450 Software Verification, Validation and Testing

**Departmental Elective- V**

1. CSE-472 Object Oriented Software Engineering  
2. CSE-474 Simulation and Modeling  
3. CSE-476 Data warehousing and Data Mining
UNIT 1.

**Introduction:** Concepts of neural networks, Characteristics of Neural Networks, Historical Perspective, and Applications of Neural Networks.

**Fundamentals of Neural Networks:** The biological prototype, Neuron concept, Single layer Neural Networks, Multi-Layer Neural Networks, terminology, Notation and representation of Neural Networks, Training of Artificial Neural Networks.

Representation of perceptron and issues, perceptron learning and training, Classification, linear Separability

Unit 2

**Hopfield nets:** Structure, training, and applications, Stability

**Back propagation:** Concept, Applications, and Back Propagation Training Algorithms.

**Counter Propagation Networks:** Kohonan Network, Grossberg Layer & Training, applications of counter propagation, Image classification.

UNIT 3

**Bi-directional Associative Memories:** Structure, retrieving a stored association, encoding associations, memory capacity.

**ART:** ART architecture, ART classification operation, ART implementation, and characteristics of ART.

Image Compression Using ART

UNIT 4

**Optical Neural Networks:** Vector Matrix Multipliers, Hop field net using Electro optical matrix multipliers, Holographic correlator, Optical Hopfield net using Volume Holograms.

**The Cognitrons and Neocognitrons:** Their structure and training.


Note: - There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

Books

1. Li Min Fu," Neural Networks in Computer Intelligence", McGraw-Hill, Inc.
Interactive Computer Graphics

CSE-404
L T P Theory: 100
4 1 - Sessional: 25

UNIT- 1


UNIT- 2

Display Description: Screen co-ordinates, user co-ordinates, Graphical data structures (compressed incremental list, vector list, use of homogeneous coordinates); Display code generation Graphical functions: the view algorithm. Two-dimensional transformation, Line drawing. Circle drawing algorithms.

UNIT- 3


UNIT-4

3-D Graphics: Wire-frame, perspective display, Perspective depth, projective transformations, Hidden line and surface elimination. Transparent solids, shading. Two dimensional Transformations. 3-dimesional Transformations. Interactive Graphical Techniques GUI.

Note: - There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

Books

5. Kelley Bootle, Mastering Turbo C
Neural Networks (Pr.)

CSE-406

L T - - P
3

Practical: 50
Sessional: 50

Design and train

1. NN for AND, OR gate using perceptron.
2. Perceptron to classify odd and even numbers.
3. NN for alphabet recognition using backpropagation.
4. Hopfield network for recognizing patterns such as ‘+’ and ‘-‘.
5. NN for EXOR classification using Back propagation.
6. CPN for image classification.
7. Name and Telephone number recognition system

Note: Atleast 5 to 10 more exercises are to be given by the teacher concerned.
Unit-1

Architecture of distributed operating system: Introduction, motivation, system architecture type, issues in distributed operating system, Communication primitive.

Unit-2


Unit-3

Distributed dead lock detection: Introduction, dead lock handling, strategies, issues in deadlock detection & resolution, Control organization, centralized, distributed & hierarchical detection algorithm.

Unit-4

Distributed file system: Introduction, architecture mechanism for building, design issues, log structured file system.

Distributed Scheduling: Introduction, motivation, issues in load distribution, component of load algorithm, stabilizing load distribution algorithm, performance comparison, selection of a suitable load sharing algorithm, requirement for load distribution, task migration, issues in task migration.

Note: There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

BOOKS

2. A S Tanenbaum : Modern Operating Systems , PHI.
Software Quality Models and Testing

CSE-442

(Departmental Elective IV)

L T P  
3 1 -  

Theory : 75  
Sessional: 50

Unit-1

Software Quality: Meaning and scope, software quality factors, software quality metrics, relationship b/w quality factors and quality metrics, quality management system, software reviews, formal technical reviews, correctness proof, statistical quality assurance, clear room, software engineering, standards of software quality assurance.

Unit-2

Software Reliability: meaning and its relation with software quality, reliability modeling-exponential failure time models (viz., Jelinski Moranda model, Schneidiwind’s model, Musa’s basic execution time model, hyperexponential model), Weibull and gamma failure time model (viz. Weibull model, S-shaped reliability growth model), and infinite failure category models (viz. Duane’s model, geometric model, Muse-Okumto model). Types of failure, bath-tub Curve, Exponential law of reliability.

Unit-3

Software Testing: Meaning. Scope and its relationship with software quality, software testing techniques: white box testing, basis path testing, control structure testing and black box testing, etc.
Software testing strategies: unit testing, integration testing, validation testing and system testing, etc.

Unit-4

Concept of repair and maintenance, concept of availability and its relation with reliability and maintainability, preventive maintenance, Software maintenance, the management of reliable software, Automatic error detection and error correction.

Note: There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.
Books


5. Concepts of Reliability by L SriNath
6. Software Reliability By K.K. Aggarwal
7. Software Reliability by H Koptez.
**Unit-1**

**Introduction to Bioinformatics:** Introduction, outline of proteins, primary structure: the 20 amino acids – chemical structure & properties; chirality, different types of side chain: relevance to mutation, size, aliphatic/aromatic, polarity, charge, hydrophobicity; disulphide bonds, molecular models, polypeptide geometry: the folding chain, nomenclature, molecular graphics, Structure evolution and mutation genetic information- the triplet code; DNA structure Synthesis of proteins: cell biology background; transcription; RNA polymerase, introns, exons, splicing translation: ribosomes, strat/stop codons, post-translational processing

**Unit-2**

**Computing evolution:** Phylogenetic Analysis Sequence- based taxonomy: overview and assumptions, from Multiple Alignment to phylogeny Neighbor, Joining Maximum Likelyhood Vs. Parsimony, The molecular Clock, Computer Tools for patterns, mapping and phylogenetic analysis, Mathematical tools of proteins and nucleic acids, sequence- Function Relationships Sequence Homology and Conserved Regions, Conserved DNA Sequences.

**Unit-3**

**Bioinformatics tools:** Networks- WWW, CERN EMBnet; EMBL Database, SEQNET, Gen Bank, NLM ,Etc. , Sequence Databases and Sequence Analysis: Genomic, CDNA EMBL database GenBank Protein sequence, Pattern recognition tools Similarity searching, secondary sources, genome databases, Molecular graphics software and other packages, To find sequences based on keywords & phrases, to grab individual sequences or whole groups of Sequences from a database

**Unit-4**

**Genomics:** Introduction, genome scale sequencing, comparative and evolutionary genomics, microarrays, proteomics, pharmacogenomics, Development using computer tools for sequencing projects, PCR and restriction mapping practical and theoretical problems in sequencing. The challenges of whole genome sequencing, web based tools for restriction mapping, new technologies and new bioinformatics tools.

Note: There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.
Books
# Expert Systems

**CSE-446**

(Practical Elective IV)

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

**Theory:** 75  
**Sessional:** 50

## Unit-1

Features of expert system, Representation and organization of knowledge, Basics characteristics, types of problems handled by expert systems, Case study of PROSPECTOR.

## Unit-2

**Expert System Tools:** Techniques of knowledge representations in expert systems, knowledge engineering, System-building aids, support facilities, stages in the development of expert systems.

## Unit-3

**Building an Expert System:** Expert system development, Selection of tool, Acquiring Knowledge, Building process.

## Unit-4

**Problems with Expert Systems:** Difficulties, common pitfalls in planning, dealing with domain expert, difficulties during development.

Note: There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

## Books

Real Time Systems and Software

CSE-448

(Departmental Elective IV)

L T P Theory: 75
3 1 - Practical: 50

Unit-1
Introduction, Real-time Versus Conventional Software, Computer Hardware for Monitoring and
Control, Software Engineering Issues.

Process and State-based Systems model, Periodic and Sporadic Process, Cyclic Executives, CE
definitions and Properties, Foreground-Background Organizations, Standard OS and
Concurrency – Architectures, Systems Objects and Object-Oriented Structures, Abstract Data
Types, General Object Classes

Unit-2
Requirements and Design Specifications: Classification of Notations, Data Flow Diagrams,
Tabular Languages, State Machine, Communicating Real Time State Machine- Basic features,
Timing and clocks, Semantics Tools and Extensions, Statecharts-Concepts and Graphical
Syntax, Semantics and Tools
Declarative Specifications: Regular Expressions and Extensions, Traditional Logics-
Propositional Logic, Predicates, Temporal logic, Real time Logic

Unit-3
Deterministic Scheduling : Assumptions and Candidate Algorithms, Basic RM and EDF Results,
Process Interactions-Prority Inversion and Inheritance

Execution Time Prediction: Measurement of Software by software, Program Analysis with
Timing Schema, Schema Concepts, Basic Blocks, Statements and Control, Schema Practice,
Prediction by optimisation, System Interference and Architectural Complexities

Unit-4
Timer Application, Properties of Real and ideal clocks, Clock Servers – Lamport’s Logical
clocks, Monotonic Clock service, A software Clock server, Clock Synchronization- Centralized
Synchronization, Distributed Synchronization

Programming Languages: Real Time Language Features, Ada-Core Language, Annex
Mechanism for Real Time Programming, Ada and Software Fault Tolerance, Java and Real-time
Extensions, CSP and Occam

Operating Systems: Real Time Functions and Sevices, OS Architectures-Real Time UNIX and
POSIX, Issues in Task management- Processes and Threads, Scheduling, Synchronization and
communication
Note: - There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

Books
1. Real – Time Systems and software by Alan C. Shaw; John Wiley & Sons Inc
Unit-1
**Introduction:** What is software testing and why it is so hard?, Error, Fault, Failure, Incident, Test Cases, Testing Process, Limitations of Testing, No absolute proof of correctness, Overview of Graph Theory & Discrete Mathematics.

**Functional Testing:** Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique.

Unit-2
**Structural Testing:** Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing.

**Reducing the number of test cases:**
Prioritization guidelines, Priority category, Scheme, Risk Analysis, Regression Testing, Slice based testing

Unit-3


**Object Oriented Testing:** Issues in Object Oriented Testing, Class Testing, GUI Testing, Object Oriented Integration and System Testing.

Unit-4


**Note:** - There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

**Books**
Departmental Elective-V
Object Oriented Software Engineering

CSE-472

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

Theory : 75
Sessional: 50

Unit-1

Design Objects, Class Hierarchy, inheritance, polymorphism, object relationships and associations, aggregations and object containment, object persistence, meta -classes, Object-oriented systems development life cycle, Software development process object oriented systems development: a use-case driven approach.

Unit-2

Object modeling techniques as software engineering methodology, Rumbaugh methodology, Jacobson methodology, Booch methodology, patterns, frameworks, the unified modeling language (UML).

Unit-3

Analysis Process, Use-Case Driven Object Oriented Analysis, Use-Case Model, Object Classification, Theory, Different Approaches for identifying classes, classes, responsibilities and Collaborators, identifying Object Relationships, attributes and Methods, super-sub Class Relationships, Apart of Relationships-Aggregation , Class Responsibilities , Object Responsibilities.

Unit-4

Object Oriented design process, corollaries, design axioms, design patterns, object oriented design philosophy, UML Object Constraint Language, Designing Classes : The Process, Class Visibility, Refining Attributes, Designing Methods and Protocols, Packages and Managing classes, Designing interface objects, View layer interface design, Macro and Micro level interface design process.

Note: There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

BOOKS

2. Rumbaugh et.al.,Object Oriented Modeling and Design, PHI, 1997
4. William Stallings: Data and Computer Communications 5/e, PHI.
Simulation and Modeling CSE-474  
(Departmental Elective- V) 

L  T  P  
3  1  -  
Theory : 75 
Sessional: 50 

Unit-1  
Introduction: System Concepts, system boundaries and environment, continuous and discrete 
systems, system modeling, types of Models, Modeling methodology, Model validation, 
Principles & Nature of Computer modeling and simulation. 

Unit-2  
Continuous and Discrete: Analog vs. Digital Simulation, Continuous simulation vs. Numerical 
Integration, Concepts of simulation of continuous and discrete system with the help of live 
example, generation of random numbers, generation of non-uniformly distributed random 
numbers, generation of Poisson and Erlang variates. 

Unit -3  
Simulators for the Live systems: Simulation of a water reservoir system, Simulation of a 
hypothetical Computer, Simulation of queuing Systems, basic concepts of queuing theory, 
simulation of single server, two server and general queuing theory, simulation in inventory 
control systems, elements of Inventory theory, inventory models, simulators for complex 
inventory systems. 

Unit-4  
Design and Evaluation of Simulation Experiments: Length of simulation, run variance 
reduction techniques. Experiment layout and Validation. 

Note: There will be 8 questions in all. Two Questions will be set from each unit. Students are 
required to attempt five questions selecting at least one question from each unit. 

Books  
3. Neelankavil Frances: Computer Simulation and Modelling, John Wiley & Sons, New 
4. Payne, James A.: Introduction to simulation: Programing Techniques and Methods of 
Data Warehousing and Data Mining

CSE-476

( Departmental Elective-V )

L T P Theory: 75
3 1 - Practical: 50

Unit-1
Data Warehousing: Definition, Scope, Practical Implications, Structures and functions.

Data Mining: Process, Technologies & Rules, platform tools & tool characteristics, operational vs. information systems.

Unit-2
Types of Data Warehouses: Host based, single stage, LAN based, Multistage, stationary distributed & virtual data-warehouses.

Unit-3
Data warehouses architecture: Metadata, operational data & operational data bases. Data warehouse architecture model, 2-tier, 3-tier & 4-tier data warehouses.

OLAP & DSS support in data warehouses.

Unit-4
Data Mining: Knowledge discovery through statistical techniques, Knowledge discovery through neural networks, Fuzzy tech. & genetic algorithms.

Note: - There will be 8 questions in all. Two Questions will be set from each unit. Students are required to attempt five questions selecting at least one question from each unit.

Books

5. “Data Mining”, A. K. Pujari; Longman Publisher