**Bachelor of Technology (Computer Science & Engineering)** w.e.f 2015 till 2020

**Scheme of Studies/Examination**

**Semester III**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **S. No.** | **Course No.** | **Subject** | **L:T:P** | **Hours/Week** | **Credits** | **Examination Schedule (Marks)** | | | | **Duration of Exam (Hrs)** |
| | | | | | | **Major Test** | **Minor Test** | **Practical** | **Total** | |
| 1 | HS-201 | Fundamentals of Management | 3:0:0 | 3 | 3.0 | 75 | 25 | 0 | 100 | 3 |
| 2 | CSE-201 | Discrete Structures | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 3 | CSE-203 | Data Structures | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 4 | CSE-205 | Data Base Management Systems | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 5 | CSE-207 | Digital Electronics | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 6 | CSE-209 | Programming Languages | 3:1:0 | 3 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 7 | CSE-211 | Data Structures Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 8 | CSE-213 | Digital Electronics Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 9 | CSE-215 | Database Management Systems Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| | | **Total** | | **31** | **25** | **450** | **270** | **180** | **900** | |
| 10 | MPC 202 | Energy Studies* | 3:0:0 | 3 | | 75 | 25 | 0 | 100 | 3 |

*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

| HS-201 | FUNDAMENTALS OF MANAGEMENT | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
| 3 | 0 | - | 3.0 | 75 | 25 | 100 | 3 |
| 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

**Evolution of Management Thought**: Development of Management Thought- Scientific management, Administrative Theory of Management, Bureaucratic Organization, Behavioral approach (Neo Classical Theory): Human Relations Movement; Behavioral Science approach; Modern approach to management – Systems approach and contingency approach.

## 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
2. Principles of Management - Koontz & O’Donnel; (McGraw Hill)

**Recommended books**

1. Business Organization and Management – Basu ; Tata McGraw Hill
2. Management and OB-- Mullins; Pearson Education
3. Essentials of Management – Koontz, Tata McGraw-Hill
4. Management Theory and Practice – Gupta, C.B; Sultan Chand and Sons, new Delhi
5. Prasad, Lallan and S.S. Gulshan. Management Principles and Practices. S. Chand & Co. Ltd., New Delhi.
6. Chhabra, T.N. Principles and Practice of Management. Dhanpat Rai & Co., Delhi.
7. Organizational behavior – Robins Stephen P; PHI.

| **CSE-201** | **Discrete Structures** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3** |
| **Purpose** | **To provide the conceptual knowledge of Discrete structure.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | **To study various fundamental concepts of Set Theory and Logics.** | | | | | | |
| **CO 2** | **To study and understand the Relations, diagraphs and lattices.** | | | | | | |
| **CO 3** | **To study the Functions and Combinatorics.** | | | | | | |
| **CO 4** | **To study the Algebraic Structures.** | | | | | | |

### Unit 1 Set Theory & Logic

Fundamentals - Sets and subsets, Venn Diagrams, Operations on sets, Laws of Set Theory, Power Sets and Products, Partition of sets, The Principle of Inclusion- Exclusion.

Logic : Propositions and Logical operations, Truth tables, Equivalence, Implications, Laws of Logic, Normal forms, Predicates and quantifiers, Mathematical Induction.

### Unit 2: Relations, diagraphs and lattices

Product sets and partitions, relations and diagraphs, paths in relations and diagraphs, properties of relations, equivalence and partially ordered relations, computer representation of relations and diagraphs, 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:**

- Elements of Discrete Mathematics C.L Liu, 1985, Reprinted 2000, McGraw Hill
- Discrete mathematical structures by B Kolman RC Busby, S Ross PHI Pvt. Ltd.
-

**Reference:**

- Discrete Mathematical Structures with Applications to Computer Science , by Tremblay J.P, and Manohar R., McGraw Hill Book Company, 1975, International Edition, 1987.
- Discrete and Combinatorial mathematics ", Ralph P., Grimaldi, Addison-Wesley Publishing Company, Reprinted in 1985.
- Discrete Mathematics and its Applications ", Kenneth H.Rosen, McGraw Hill Book Company, 1999. Sections: 7.1 to 7.5.
- Discrete Mathematics for computer scientists and Mathematicians, Joe L. Mott, Abraham

| CSE-203 | Data Structures | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hour** |
| **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 operations'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.** | | | | | | |

## 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

**Stacks**: Definition, Implementation of Stacks and Its Operations, Evaluation of Infix, prefix and Postfix Expression, Inter-conversion of Infix Expression, Prefix and Post-Fix Expression, Implementation of Merge Sort and Quick Sort Algorithm.

**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

**Linked Lists**: Dynamic Implementations, Need of Dynamic Data Structures, Single Link List and Its Dynamic Implementation, Traversing, Insertion, Deletion Operations on Single Link Lists. Comparison between Static and Dynamic, Implementation of Linked List. Dynamic Implementation of Stacks and Queues.

Circular Link Lists and Doubly Link List, Dynamic Implementation of Primitive Operations on Doubly Linked Lists and Circular Link List.

## 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.

Introduction to Binary Search Trees: B trees, B+ trees, AVL Trees, Threaded Binary trees, Balanced Multi-way search trees, Implementation of Heap Sort Algorithm.

**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

**References**:

- Shukla, Data Structures using C++, Wiley India
- Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original edition, Addison-Wesley, 1999, Low Priced Edition.
- 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
- Introduction to Computers Science -An algorithms approach , Jean Paul Tremblay, Richard B. Bunt, 2002, T.M.H.
- Data Structure and the Standard Template library – Willam J. Collins, 2003, T.M.H

| **CSE-205** | **DATA BASE MANAGEMENT SYSTEMS** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hour** |
| **Purpose** | **To familiarize the students with Data Base Management system** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | To provide introduction to relational model. | | | | | | |
| **CO 2** | To learn about ER diagrams. | | | | | | |
| **CO 3** | To understand about Query Processing and Transaction Processing. | | | | | | |
| **CO 4** | To understand about the concept of functional dependencies. | | | | | | |
| **CO 5** | To learn the concept of failure recovery. | | | | | | |
| **CO 6** | To understand the concurrency control. | | | | | | |

## UNIT I

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

**Entity-Relationship Model:** Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features.

## 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**

Issues and Models for Resilient Operation -Undo/Redo Logging-Protecting against Media Failures.

**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. Ramez Elmasri , Shamkant B. Navathe ,”Fundamentals of Database systems”, Pearson
2. 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

| **CSE-207** | **Digital Electronics** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hour** |
| **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

Design procedure - Half adder, Full Adder, Half subtractor, Full subtractor , Parallel binary adder, parallel binary Subtractor, Fast Adder, Carry Look Ahead adder, Serial Adder/Subtractor, BCD adder, Binary Multiplier, Binary Divider, Multiplexer/ De-multiplexer, decoder, encoder, parity checker, parity generators, code converters, Magnitude Comparator.

## UNIT III SEQUENTIAL CIRCUITS

Latches, Flip-flops - SR, JK, D, T, and Master-Slave - Characteristic table and equation, Application table, Edge triggering, Level Triggering, Realization of one flip-flop using other flip-flops, serial adder/subtractor, Asynchronous Ripple or serial counter, Asynchronous Up/Down counter, Synchronous counters, Synchronous Up/Down counters, Programmable counters, Design of Synchronous counters: state diagram, State table, State minimization, State assignment, Excitation table and maps-Circuit implementation, Modulo-n counter, 555 Timer, Registers - shift registers, Universal shift registers, Shift register counters, Ring counter, Shift counters, Sequence generators.

## 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, MOSFET 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**

**1.** Donald P. Leach and Albert Paul Malvino, Digital Principles and Applications, 8th Edition, TMH, 2003.M.

2. Morris Mano, Digital Design, 3rd Edition, Prentice Hall of India Pvt. Ltd., 2003 / Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2003.

**REFERENCES**

**1.** A.K. Maini, Digital Electronics, Wiley India

2. John F. Wakerly, Digital Design, Fourth Edition, Pearson/PHI, 2006

2. John. M Yarbrough, Digital Logic Applications and Design, Thomson Learning, 2002.

3. S. Salivahanan and S. Arivazhagan, Digital Circuits and Design, 3rd Edition., Vikas Publishing House Pvt. Ltd, New Delhi, 2006

4. William H. Gothmann, Digital Electronics, 2nd Edition, PHI, 1982.

5. Thomas L. Floyd, Digital Fundamentals, 8th Edition, Pearson Education Inc, New Delhi, 2003

6. Donald D. Givone, Digital Principles and Design, TMH, 2003.

| **CSE-209** | **Programming Languages** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hour** |
| **Purpose** | To introduce the principles and paradigms of programming languages for design and implement the software intensive systems. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO 1** | To introduce the basic concepts of programming language, the general problems and methods related to syntax & semantics. | | | | | | |
| **CO 2** | To introduce the structured data objects, subprograms and programmer defined data types. | | | | | | |
| **CO 3** | To outline the sequence control and data control. | | | | | | |
| **CO 4** | To introduce the concepts of storage management using programming languages. | | | | | | |

### 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.
2. Allen Tucker & Robert Noonan, Programming Languages–Principles and Paradigms, Tata McGraw-Hill, 2009.

**Reference Books:**

1. Ellis Horowitz, Fundamentals of Programming Languages, Galgotia Publications, 2010.
2. C. Ghezzi, Programming Languages Concepts, Wiley Publications, 2010.

| **CSE-211** | **Data Structures Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **3** | **1.5** | **40** | **60** | **100** | **3** |
| **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 serach 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 | Digital Electronics Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **3** | **1.5** | **40** | **60** | **100** | **3** |
| **Purpose** | **To learn the basic methods for the design of digital circuits and systems.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | **To Familiarization with Digital Trainer Kit and associated equipment.** | | | | | | |
| **CO 2** | **To Study and design of TTL gates** | | | | | | |
| **CO 3** | **To learn the formal procedures for the analysis and design of combinational circuits.** | | | | | | |
| **CO 4** | **To learn the formal procedures for the analysis and design of sequential circuits** | | | | | | |

## 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.
3. Design and realize a given function using K-Maps and verify its performance.
4. To verify the operation of Multiplexer and De-multiplexer.
5. To verify the operation of Comparator.
6. To verify the truth table of S-R, J-K, T, D Flip-flops.
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.
12. Study of Encoder and Decoder.
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.**

| **CSE-215** | **DATABASE MANAGEMENT SYSTEMS LAB** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **-** | **-** | **3** | **1.5** | **40** | **60** | **100** | **3** |
| **Purpose** | **To familiarize the students with the basics of Operating Systems** | | | | | | |
| **Course Outcomes** | | | | | | | |
| CO1 | To understand basic DDL commands | | | | | | |
| CO 2 | To learn about DML and DCL commands | | | | | | |
| CO 3 | To understand the sql queries using SQL operators | | | | | | |
| CO 4 | To understand the concept of relational algebra | | | | | | |
| CO5 | To learn various queries using date and group functions | | | | | | |
| CO6 | To understand the nested queries | | | | | | |
| CO7 | To learn view, cursors and triggers. | | | | | | |

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 | ENERGY STUDIES | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
| 3 | - | - | 0 | 75 | 25 | 100 | 3 |
| Purpose | To make the students conversant with the basics concepts and conversion of various form of Energy | | | | | | |
| Course Outcomes | | | | | | | |
| CO1 | An overview about Energy , Energy Management, Audit and tariffs | | | | | | |
| CO2 | Understand the Layout and working of Conventional Power Plants | | | | | | |
| CO3 | Understand the Layout and working of Non-Conventional Power Plants | | | | | | |
| CO4 | To understand the role of Energy in Economic development and Energy Scenario in India | | | | | | |

## UNIT-I

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

**Energy Management:** General Principles of Energy Management, Energy Management Strategy.

**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.
2. Non-conventional energy resources- Shobhnath Singh, Pearson.
3. Soni,Gupta,Bhatnagar: Electrical Power Systems – DhanpatRai& Sons
4. NEDCAP: Non Conventional Energy Guide Lines
5. G.D. Roy :Non conventional energy sources
6. B H Khan :Non Conventional energy resources - McGraw Hill
7. Meinel A B and Meinal M P,Addison:Applied Solar Energy- Wesley Publications
7. George Sutton: Direct Energy Conversion -McGraw

**Bachelor of Technology (Computer Science & Engineering)**

**Scheme of Studies/Examination**

**Semester IV**

| S. No. | Course No. | Subject | L:T:P | Hours/ Week | Credits | Examination Schedule (Marks) | | | | Duration of Exam (Hrs) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Major Test | Minor Test | Practical | Total | |
| 1 | AS-201 | Mathematics-III | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 2 | CSE-202 | Object Oriented Programming | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 3 | CSE-204 | Internet Fundamental | 3:0:0 | 3 | 3.0 | 75 | 25 | 0 | 100 | 3 |
| 4 | CSE-206 | Digital Data Communication | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 5 | CSE-208 | Microprocessor & Interfacing | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 6 | CSE-210 | Operating System | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 7 | CSE-212 | Object Oriented Programming Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 8 | CSE-214 | Microprocessor Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 9 | CSE-216 | Internet Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| | | **Total** | | **32** | **25.0** | **450** | **270** | **180** | **900** | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | MPC 201 | Enviornment Studies* | 3:0:0 | 3 | | 75 | 25 | | 100 | 3 |

*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 | Mathematics-III | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
| 3 | 1 | - | 3.5 | 75 | 25 | 100 | 3 |
| Purpose | To provide the conceptual knowledge of Engineering mathematics | | | | | | |
| Course Outcomes | | | | | | | |
| CO 1 | To study various fundamental concepts of Fourier series and Fourier Transformation. | | | | | | |
| CO 2 | To study and understand the functions of a complex variables. | | | | | | |
| CO 3 | To study the Probability Distributions. | | | | | | |
| CO 4 | To study the linear programming problem formulation. | | | | | | |

## 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.

Fourier Transforms: Fourier integrals, Fourier transforms, Fourier cosine and sine transforms. Properties of Fourier transforms, Convolution theorem, Perseval’s identity, Relation between Fourier and Laplace transforms, Fourier transforms of the derivatives of a function, Application to boundary value problems.

## 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**

1. Higher Engg. Mathematics: B.S. Grewal
2. Advanced Engg. Mathematics: E. Kreyzig

**Reference Book**

1. Complex variables and Applications: R.V. Churchil; Mc. Graw Hill
2. Engg. Mathematics Vol. II: S.S. Sastry; Prentice Hall of India.
3. Operation Research: H.A. Taha
4. Probability and statistics for Engineer: Johnson. PHI.

| CSE-202 | Object Oriented Programming | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hour** |
| **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** | | | | | | |

## Unit–1

Introduction to C++, C++ Standard Library, Illustrative Simple C++ Programs. Header Files, Namespaces, Application of object oriented programming.

Object Oriented Concepts, Introduction to Objects and Object Oriented Programming, Encapsulation, Polymorphism, Overloading, Inheritance, Abstract Classes, Accessifier (public/ protected/ private), Class Scope and Accessing Class Members, Controlling Access Function, Constant, Class Member, Structure and Class

## 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

Polymorphism, Pointer to Derived class, Virtual Functions, Pure Virtual Function, Abstract Base Classes, Static and Dynamic Binding, Virtual Deconstructors.

Fundamentals of Operator Overloading, Rules for Operators Overloading, Implementation of Operator Overloading Like <<,>> Unary Operators, Binary Operators.

## Unit-4

Text Streams and binary stream, Sequential and Random Access File, Stream Input/ Output Classes, Stream Manipulators.

Basics of C++ Exception Handling, Try, Throw, Catch, multiple catch, Re-throwing an Exception, Exception specifications.

Templates: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non-Type Template arguments.

**Text Books**:

- Object Oriented Programming in Turbo C++ by Robert Lafore, 1994, The WAITE Group Press.
- The complete reference C ++ by Herbert shieldt Tata McGraw Hill

**References Books**

- Shukla, Object Oriented Programming in c++, wiley india
- C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall
- Programming with C++ By D Ravichandran, 2003, T.M.H

| CSE-204 | Internet Fundamentals | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **0** | **-** | **3** | **75** | **25** | **100** | **3** |
| **Purpose** | **To provide the conceptual knowledge of Internet and methodologies used in web and secure internet communication and networking.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | **To study various fundamental concepts of Internetworking techniques with their characteristics.** | | | | | | |
| **CO 2** | **To study and understand the requirements for world-wide-web formats and techniques.** | | | | | | |
| **CO 3** | **To study the E-mail functioning and basics of HTML, XML and DHTML languages.** | | | | | | |
| **CO 4** | **To study the functioning of Servers and Privacy and Security related mechanisms.** | | | | | | |

## 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

Introduction, Miscellaneous Web Browser details, searching the www: Directories search engines and meta search engines, search fundamentals, search strategies, working of the search engines, Telnet and FTP, HTTP, Gophar Commands, TCP/IP. Introduction to Browser, Coast-to-coast surfing, hypertext markup language, Web page installation, Web page setup, Basics of HTML & formatting and hyperlink creation.Using FrontPage Express, Plug-ins.

## UNIT-III : INTERNET PLATEFORM 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
- Networking Essentials – Firewall Media.Latest-2015
- www.secinf.com
- www.hackers.com
- Alfred Glkossbrenner-Internet 101 Computing MGH, 2013

| **CSE-206** | **Digital Data Communication** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3** |
| **Purpose** | **To provide the conceptual knowledge of data preparation and signal transmission methodologies used in data communication and networking.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | **To study various analog communication techniques and with their characteristics.** | | | | | | |
| **CO 2** | **To study and understand the requirements for analog/digital data to analog/digital signal conversion techniques.** | | | | | | |
| **CO 3** | **To study the error and flow control techniques in communication and networking.** | | | | | | |
| **CO 4** | **To study the concept of multiplexing and applied multiple access techniques specially in satellite communication.** | | | | | | |

## UNIT-1

### 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

Digital data, Digital signals: Encoding schemes: NRZ-L, NRZ-I, Manchester-Diff-Manchester-encoding,Pseudoternary-Bipolar-AMI,B8ZS- HDB3 – Evaluation factors-Digital data, analog signals: Encoding Techniques –ASK-FSK-PSK-QPSK-Performance comparison-Analog data, digital signals: Quantization-Sampling theorem-PCM-Delta modulation-Errors- comparison- Analog Data, analog signals: Need for modulation -0 Modulation methods – Amplitude modulation- Angle modulation- Comparison.

## UNIT-III

### DIGITAL DATA COMMUNICATION TECHNIQUES

Asynchronous and synchronous transmission –Error Detection techniques: Parity checks – Cycle redundancy checks-Checksum-Error Correcting codes: Forwards and backward error corrections, Transmission media. Communication Topologies.

DTE & DCE interface: Characteristics of DTE-DCE interface. Interfaces: Rs-232-C, Rs-449/422, A/423-A.

## 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

1. Forouzen, “Data Communication & Networking”,Tata Mcgraw Hill
2. Proakin, “Digital Communications”,Mc Graw Hill.
3. W. Stalling, “Wireless Communication and Networks” Pearson.

### REFERENCES

1. Stallings, “Data & computer Communications”,PHI.
2. Roden, “Digital & Data Communication Systems”,PHI.
3. Irvine, Data communications & Networks An engineering approach, wiley india

| CSE-208 | Microprocessor & Interfacing | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3** |
| **Purpose** | **To learn the architecture and programming of Intel family microprocessors and its interfacing.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | **To study the Architecture of 8085 microprocessors** | | | | | | |
| **CO 2** | **To learn the architecture 8086 Microprocessor and its interfacing to memories** | | | | | | |
| **CO 3** | **To learn the instruction set of 8086 Microprocessor and assembly language programming of 8086 Microprocessor.** | | | | | | |
| **CO 4** | **To learn interfacing of interrupts, basic I/O and DMA with 8086 Microprocessor** | | | | | | |

## 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.

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

**Text Books:**

- Barry B. Brey, "The Intel Microprocessor8086/8088, 80186", Pearson Education, Eighth Edition, 2009
- D.V. Hall, Microprocessors and Interfacing, McGraw Hill 2nd ed.

**Reference Books:**

- Liu, Gibson, "Microcomputer Systems: The 8086/88 Family", 2nd Edition, PHI,2005
- Kenneth Ayala, "The 8086 Microprocessor: Programming & Interfacing the PC", Cengage Learning, Indian Edition, 2008
- Kip Irvine, "Assembly language for IBM PC", PHI, 2nd Edition, 1993
- Peter Abel, "Assembly language programming", Pearson Edu,5th Edition,2002
- Uffenback, "The 8086 Family Design" PHI, 2nd Edition.
- Walter A Triebel and Avtar Singh; The 8088 and 8086 Microprocessors, Programming, Interfacing, Software, Hardware and Applications, Fourth Edition, Pearson Education.

| **CSE-210** | **OPERATING SYSTEMS** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3** |
| **Purpose** | **To familiarize the students with the basics of Operating Systems** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | **To understand the structure and functions of Operating system.** | | | | | | |
| **CO 2** | **To learn about processes, threads and scheduling algorithms.** | | | | | | |
| **CO 3** | **To understand the principle of concurrency.** | | | | | | |
| **CO 4** | **To understand the concept of deadlocks.** | | | | | | |
| **CO5** | **To learn various memory management schemes.** | | | | | | |
| **CO6** | **To study I/O management and file systems.** | | | | | | |
| **CO7** | **To study the concept of protection and security.** | | | | | | |

## UNIT 1

**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:**

1. Operating Systems : Internals and Design Principles, William Stallings, Pearson
2. Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Wiley

**Reference books:**

1. Operating systems: a concept based approach", Dhananjay M. Dhamdhere, McGraw Hill .
2. Operating Systems Design and Implementation" ,(Prentice Hall Software Series) Andrew S Tanenbaum and Albert S Woodhull
3. Taub & Schilling, Principles of Communication Systems, TMH.
4. Mithal G K, Radio Engineering, Khanna Pub.
5. Sirnon Haykin, Communication Systems, John Wiley

| CSE-212 | Object Oriented Programming Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Minor Test | Practical | Total | Time |
| 0 | 0 | 3 | 1.5 | 40 | 60 | 100 | 3 Hour |
| 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 cenitmetres 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<< end1:}
};
```

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 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 **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 kay 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 baseclass 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 = ½ * x * y

| **CSE-214** | **Microprocessor Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **3** | **1.5** | **40** | **60** | **100** | **3 Hour** |
| **Purpose** | **Write the efficient Assembly Language Program for different problem statements and implement different system interfacing.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO 1** | **Understanding different steps to develop program such as Problem definition, Analysis, Design of logic, Coding, Testing, Maintenance (Modifications, error corrections, making changes etc.)** | | | | | | |
| **CO 2** | **To be able to apply different logics to solve given problem.** | | | | | | |
| **CO 3** | **To be able to write program using different implementations for the same problem** | | | | | | |
| **CO 4** | **Use of programming language constructs in program implementation** | | | | | | |

**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 | Internet Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Minor Test | Practical | Total | Time |
| 0 | 0 | 3 | 1.5 | 40 | 60 | 100 | 3 Hour |
| 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. | | | | | | |

**PC Software**: Application of basics of MS Word 2000, MS Excel 2000, MS Power Point 2000, MS Access 2000, HTML

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.
4. Design Web pages containing information of the Deptt.

**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.

| MPC-201 | ENVIRONMENTAL STUDIES | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **-** | **-** | **-** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **To learn the multidisciplinary nature, scope and importance of Environmental Studies** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | **Basic concepts of Various kinds of Microscopy and Centrifugation Techniques** | | | | | | |
| **CO2** | **To learn the theoretical and practical aspects of Electrophoresis and Chromatography Techniques** | | | | | | |
| **CO3** | **To learn the concepts of different kinds of Spectroscopy and Colourimetry** | | | | | | |
| **CO4** | **To understand the concept of radioisotope techniques and their applications in research** | | | | | | |

## UNIT 1

The multidisciplinary nature of environmental studies. Definition, Scope and Importance. Need for public awareness. Natural Resources: Renewable and Non-Renewable Resources: Natural resources and associated problems.

(a) Forest Resources: Use and over-exploitation, deforestation, case studies. Timber eztraction, 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

Biodiversity and its conservation. Introduction, Definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity of global, National and local levels. India as a mega-diversity nation Hot spots of Biodiversity. Threats to biodiversity: Habitat loss, poaching of wild life, man-wildlife conflicts. Endangered and endemic species of India. Conservation of Biodiversity- In situ and Ex-Situ conservation of biodiversity.

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

Social Issues and the Environment. From unsustainable to sustainable development. Urban problems related to energy. Water conservation, rain water harvesting, watershed management. Resettlement and rehabilitation of people: Its problems and concerns. Case Studies. Environmental ethics-issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies. Wasteland Reclamation. Consumerism and waste products. Environment Protection Act. Air (Prevention and Control of Pollution) Act. Water (Prevention and Control of Pollution) Act. Wildlife Protection Act. Forest Conservation Act. Issues involved in enforcement of environmental legislation. Public Awareness. Human population and the Environment. Population growth, variation among nations. Population explosion-Family Welfare Programme. Environment and human health. Human rights. Value Education. HIV/AIDS, Women and Child Welfare. Role of Information Technology in Environment and Human Health. Case Studies.

**Text Books**

1. Environmental Studies- Deswal and Deswal. Dhanpat Rai & Co.
2. Environmental Science & Engineering Anandan, P. and Kumaravelan, R. 2009. Scitech Publications (India) Pvt. Ltd., India
3. Environmental Studies. Daniels Ranjit R. J. and Krishnaswamy. 2013. Wiley India.
4. Environmental Science- Botkin and Keller. 2012. Wiley , India

# Bachelor of Technology (Computer Science & Engineering) w.e.f 2015 till 2020
## Scheme of Studies/Examination
## Semester-V

| **S. No.** | **Course No.** | **Subject** | **L:T:P** | **Hours/ Week** | **Credits** | **Examination Schedule (Marks)** | | | | **Duration of Exam (Hrs)** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | **Major Test** | **Minor Test** | **Practical** | **Total** | |
| 1 | CSE 301 | Automata Theory | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 2 | CSE 303 | Computer Networks | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 3 | CSE 305 | Design and Analysis of algorithms | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 4 | CSE 307 | Computer organisation and Architecture | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 5 | CSE 309 | Simulation &Modelling | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 6 | CSE 311 | Computer Networks Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 7 | CSE 313 | Design and Analysis of algorithms Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 8 | CSE 315 | Simulation Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 9 | CSE 317 | Seminar/Industrial Training* | 0:0:2 | 2 | 1 | 0 | 40 | 60 | 100 | |
| 10 | CSE 319 | Technical Communication and Soft Skills Lab | 0:0:2 | 2 | 1 | 0 | 100 | 0 | 100 | 3 |
| | | **Total** | | **33** | **24.0** | **375** | **385** | **240** | **1000** | |

*** Seminar/Industrial Training based on 4-6 weeks hand on training which was done after IVth Semester Exams**

| **CSE-301** | **Automata Theory** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | To understand the challenges for Theoretical Computer Science and its contribution to other sciences | | | | | | |
| **Course Outcomes(CO)** | | | | | | | |
| **CO1** | Students are able to explain and manipulate the different fundamental concepts in automata theory and formal languages. | | | | | | |
| **CO2** | Simplify automata and context-free grammars; Prove properties of languages, grammars and automata with rigorously formal mathematical methods, minimization. | | | | | | |
| **CO3** | Differentiate and manipulate formal descriptions of push down automata, its applications and transducer machines. | | | | | | |
| **CO4** | To understand basic properties of Turing machines and computing with Turing machine,the concepts of tractability and decidability. | | | | | | |

## Unit - I

**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

**Context free Grammars and Languages:** Parse Trees, Context Sensitive Grammar, Applications of Context Free Grammars, Regular Grammar, Ambiguity in Grammars and Languages. Normal forms of context free grammars, Subfamilies of Context Free Languages (CFL), Closure Properties of CFL, Chomsky Theorem, Chomsky Hierarchy, Chomsky Normal Form, Greibach Normal Form.

**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

1. J.E.Hopcroft, R.Motwani and J.D.Ullman , "Introduction to Automata Theory Languages and computation", Pearson Education Asia , 2001.
2. K.Krithivasan and R.Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education, 2009.

### References

1. Peter Linz, "An Introduction to Formal Language and Automata", 4th Edition, Narosa Publishing house , 2006.
2. M.Sipser; Introduction to the Theory of Computation; Singapore: Brooks/Cole, Thomson Learning, 1997.
3. John.C.martin, "Introduction to the Languages and the Theory of Computation",Third edition, Tata McGrawHill, 2003.

| **CSE-303** | **Computer Networks** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | To introduce the architecture and layers of computer network, protocols used at different layers. | | | | | | |
| **Course Outcomes(CO)** | | | | | | | |
| **CO1** | To understand the basic concept of networking, types, networking topologies and layered architecture. | | | | | | |
| **CO2** | To understand data link layer and MAC sub-layer` | | | | | | |
| **CO3** | To understand the network Layer functioning | | | | | | |
| **CO4** | To understand the transport layer and application layer operation | | | | | | |

### Unit -1

**Introduction**: introduction to Computer Networks, Data Communication System and its components, Data Flow, Computer network and its goals, Types of computer networks: LAN, MAN, WAN, Wireless and Wired networks, broadcast and point-to-point networks, Network topologies, protocols, interfaces and services, ISO-OSI reference model, TCP/IP architecture.

**Physical Layer**: Concept of Analog & Digital Signal, Bandwidth, Transmission Impairments: Attenuation, Distortion, Noise, Multiplexing : Frequency Division, Time Division, Wavelength Division, Introduction to **Transmission Media** : Twisted pair, Coaxial cable, Fiber optics, Wireless transmission (radio, microwave, infrared), Switching: Circuit Switching, Message Switching ,Packet Switching & comparisons, narrowband ISDN, broadband ISDN and ATM.

### 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,

**Security**: Cryptography, user authentication, security protocols in internet, public key encryption algorithm, digital signatures

**TEXT BOOK**

1. Behrouz A. Forouzan, “Data communication and Networking”, Tata McGraw Hill, Fourth Edition, 2011.
2. Computer Networks, 4th Edition, Pearson Education by Andrew S. Tanenbaum

**REFERENCES**

1. Larry L.Peterson, Peter S. Davie, “Computer Networks”, Elsevier, Fifth Edition, 2012.
2. William Stallings, “Data and Computer Communication”, Eighth Edition,Pearson Education, 2007.
3. James F. Kurose, Keith W. Ross, “Computer Networking: A Top–Down Approach Featuring the Internet”, Pearson Education, 2005.

| CSE-305 | Design and Analysis of Algorithms | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
| 3 | 1 | - | 3.5 | 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
2. Harsh Bhaisn, Algorithms: Design And Analysis Oxford University Press,2015.

### Reference Books

1. Aho, Hopcroft and Ullman : The Design and Analyses of Computer Algorithms. Addison Wesley.
2. R.B.Patel, Expert Data Structures with C, Khanna Publications , Delhi, India, 2ndEdition 2004, ISBN 81-87325-07-0, pp.1-909.
3. R.B.Patel& M.M.S Rauthan, Expert Data Structures with C++, Khana Publications, Delhi , India, 2ndEdition 2004,ISBN : 87522-03-8.
4. Horowitz, Ellis and Sahni, Sartaj : Fundamentals of Computer Algorithms, Galgotia Publications

| CSE-307 | Computer Organization and Architecture | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hrs.** |
| **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)** | | | | | | | |
| **CO1** | 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 | | | | | | |
| **CO2** | Be familiar with the design trade-offs in designing and constructing a computer processor. | | | | | | |
| **CO3** | Be familiar with the CPU design including the RISC/CISC architectures. | | | | | | |
| **CO4** | Be familiar with the basic knowledge of I/O devices and interfacing of I/O devices with computer. | | | | | | |

### 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

**Input-output organization:** I/O interface. I/O Bus and interface modules, I/O versus Memory Bus. Asynchronous data transfer: Strobe control, Handshaking, Asynchronous serial transfer. Modes of Transfer: Programmed I/O, Interrupt driven I/O, Priority interrupt; Daisy chaining, Parallel Priority interrupt.Direct memory Access, DMA controller and transfer. Input output Processor , CPU-IOP communication, I/O channel.

**TEXT BOOK:**

1. William Stallings, "Computer Organization and Architecture – Designing for Performance", Sixth Edition, Pearson Education, 2003.
2. Morris Mano, M., "Computer System Architecture," 3/e, Pearson Education, 2005.
3. John P. Hayes, "Computer Architecture and Organization," 3/e, TMH, 1998.

**REFERENCES**:

1. David A. Patterson and John L. Hennessy, "Computer Organization and Design: The Hardware/Software interface", Third Edition, Elsevier, 2005.
3. V.P. Heuring, H.F. Jordan, "Computer Systems Design and Architecture", Second Edition, Pearson Education, 2004.
4. Carl Hamacher, ZvonkoVranesic and SafwatZaky, "Computer Organization", Fifth Edition, Tata McGraw Hill, 2002.

| CSE 309 | Simulation and Modeling |
|---|---|

| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
|---|---|---|---|---|---|---|---|
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hour** |
| **Purpose** | To introduce the principles and paradigms of Computer Modeling and Simulation for solving a wide variety of problems. In addition, how to use simulator to simulate the live systems. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Learn the basic concepts of System, SystemModeling, types of Models, simulation, and need of simulation. | | | | | | |
| **CO2** | Learn the simulation of continuous and discrete systems with the help of different examples. | | | | | | |
| **CO3** | Learn the concept of generation of uniformly and non-uniformly distributed random numbers. | | | | | | |
| **CO4** | Learn the simulation of queuing system and PERT. | | | | | | |

### Unit-1

**Modeling:**System Concepts,system boundariesand environment,continuousanddiscrete systems,system modeling, typesofModels, 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 queuing Systems: basic concepts of queuing theory, simulation of single server, two server and more general queuing 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: V**ariance reduction techniques. Experiment layout and Validation.

**Case Study**: SciLab, Octave.

**Text Books:**

1.Gordon G.: Systemsimulation, Prentice-Hall of India Pvt. Ltd. New Delhi 1993

2.NarsinghDeo: SystemSimulation with Digital Computer:, PHI New Delhi, 1993

**Reference books:**

1. Neelankavil Frances: Computer Simulation and Modelling, John Wiley & Sons, NewYork, 1987.
2. Payne, James A.: Introduction to simulation: Programming TechniquesandMethodsof Analysis, McGraw-Hill International Editions, Computer Science services, New York (1998).

3.ReitamJulian: ComputerSimulation Experiments, Wiley Interscience 1971.

| **CSE-311** | **Computer Networks Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **--** | **--** | **3** | **1.5** | **40** | **60** | **100** | **3 Hour** |
| **Purpose** | **To explore networking concepts using Java programming & networking tools.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Do Problem Solving using algorithms. | | | | | | |
| **CO2** | Design and test simple programs to implement networking concepts using Java. | | | | | | |
| **CO3** | Document artifacts using applied addressing & quality standards. | | | | | | |
| **CO4** | 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.
3. Study of TCP/UDP performance.
4. Performance comparison of MAC protocols
5. Performance comparison of routing protocols.
6. Write a program:
   a. To implement echo server and client in java using TCP sockets.
   b. To implement date server and client in java using TCP sockets.
   c. To implement a chat server and client in java using TCP sockets.
7. Write a program:
   a. To implement echo server and client in java using UDP sockets
   b. To implement a chat server and client in java using UDP sockets.
   c. 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.

| **CSE-313** | **Design and Analysis of algorithms Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **--** | **--** | **3** | **1.5** | **40** | **60** | **100** | **3 Hour** |
| **Purpose** | The student willlearn 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 = {sl,s2,.....,sn} 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.
10. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
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.
16. Implement Huffman code using Greedy approach.

| **CSE 315** | **Simulation lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **-** | **-** | **3** | **1.5** | **40** | **60** | **100** | **3 Hour** |
| **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)** | | | | | | | |
| **CO1** | Learn the simulation of continuous and discrete systems with the help of different examples. | | | | | | |
| **CO2** | Learnthe concept of generation of uniformly and non-uniformly distributed random numbers. | | | | | | |
| **CO3** | Learnthe simulation of queuing system. | | | | | | |
| **CO4** | Learnthe concept of simulation CPM and PERT. | | | | | | |
| **CO5** | Learnthe concept of simulation of inventory control system. | | | | | | |

**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 toDraw graph of sine wave with respect to the time.

3: Write a program to solve following differential equation

$dy/dt = -\exp(-t)\times y^2$ by using any simulation technique.

4: Write a program to solve following differential equation by using 4th order Runge-Kutta method

$dy/dx = -2x-y$ , 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 re-order combination (P,Q) which yields the highest service level for a given value of average stock.

| **CSE-319** | **Technical Communication and Soft Skills Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **-** | **-** | **2** | **1** | **0** | **100** | **100** | **3 Hours** |
| **Purpose** | To enhance the students' oral communication skills in English | | | | | | |
| **Course Outcomes(CO)** | | | | | | | |
| **CO1** | Develop oral communicative competence in English | | | | | | |
| **CO2** | Improve fluency in English and thereby respond confidently due to reduced communication apprehension | | | | | | |
| **CO3** | Identify and explain the biological and physiological characteristic of proper voice and diction production | | | | | | |
| **CO4** | Develop correct and better pronunciation through stress on word accent, intonation, and weak forms | | | | | | |
| **CO5** | Participate in Group Discussions effectively | | | | | | |
| **CO6** | Make effective oral presentations in English | | | | | | |

## 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

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

| S. No. | Course No. | Subject | L:T:P | Hours/ Week | Credits | Examination Schedule (Marks) | | | | Duration of Exam (Hrs) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Major Test | Minor Test | Practical | Total | |
| 1 | CSE 302 | Compiler Design | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 2 | CSE 304 | Essential of Information Technology | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 3 | CSE 306 | Mobile Computing | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 4 | CSE 308 | Web Technology | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 5 | CSE 310 | Software Engineering | 3:1:0 | 4 | 3.5 | 75 | 25 | 0 | 100 | 3 |
| 6 | HS 303 | Business Intelligence and Entrepreneurship | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 7 | CSE 312 | Web Technology Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 8 | CSE 314 | Essential of Information Technology Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| 9 | CSE 316 | Software Engineering Lab | 0:0:3 | 3 | 1.5 | 0 | 40 | 60 | 100 | 3 |
| | | **Total** | | **33** | **26** | **450** | **270** | **180** | **900** | |

| **CSE-307** | **Compiler Design** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3 Hrs.** |
| **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. | | | | | | |
| **CO4** | To understand optimization of codes and runtime environment | | | | | | |

## 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 1V

### Code Optimization and Run Time Environments

Principal Sources of Optimization, Optimization of Basic Blocks, Peephole Optimization, Introduction to Global Data Flow Analysis, Source Language issues, Storage Organization, Static Storage Management, Heap Storage management, Access to non-Local Names, Parameter Passing.

### TEXT BOOK

1. Alfred Aho, Ravi Sethi, Jeffrey D Ullman, “Compilers Principles, Techniques and Tools”, Pearson Education Asia, 2003.

### REFERENCES

1. Allen I. Holub “Compiler Design in C”, Prentice Hall of India, 2003.
2. C. N. Fischer and R. J. LeBlanc, “Crafting a compiler with C”, Benjamin Cummings, 2003.
3. J.P. Bennet, “Introduction to Compiler Techniques”, Second Edition, Tata McGraw-Hill, 2003.
4. HenkAlblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001.
5. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, Thompson Learning, 2003

| CSE-304 | Essentials of Information Technology | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **0** | **3.5** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **To introduce the concepts of Object Oriented Programming using Java and RDBMS** | | | | | | |
| **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 | | | | | | |

## 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**

1. **Java™**: The **Complete Reference**,. Seventh Edition. Herbert Schildt
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**

1. Fundamentals of Database Systems, with E-book (3rd Edition) by Shamkant B. Navathe, RamezElmasri, Published January 15th 2002 by Addison Wesley Longman
2. MySQL by Paul DuBoisNew Riders Publishing
3. Murach's MySQL Paperback – 2012, by Joel Murach , Publisher: Shroff/Murach (2012)
4. SQL: The Complete Reference by James R. Groff, Paul N. Weinberg, Published March 1999 by McGraw-Hill Companies
5. Schaum's Outline of Fundamentals of Relational Databases by Ramon Mata-Toledo, Published November 15th 2000 by McGraw-Hill

| **CSE-306** | **Mobile Computing** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **0** | **3.5** | **75** | **25** | **100** | **3 Hrs.** |
| **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

Wireless Networking, Wireless LAN Overview: MAC issues, IEEE 802.11, Blue Tooth, Wireless multiple access protocols, TCP over wireless, Wireless applications, data broadcasting, Mobile IP, WAP : Architecture, Traditional TCP, Classical TCP, improvements in WAP, WAP applications.

## UNIT – III

Data management issues, data replication for mobile computers, adaptive clustering for mobilewireless 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:**

1. Rajkamal, Mobile Computing, 2/E Oxford University Press,2011.
2. J. Schiller, Mobile Communications, Addison Wesley
3. Yi Bing Lin, Wireless and Mobile Networks Architecture , John Wiley.

**Reference Books**

1. A. Mehrotra , GSM System Engineering.
2. M. V. D. Heijden, M. Taylor, Understanding WAP, Artech House.
3. Charles Perkins, Mobile IP, Addison Wesley.
4. Charles Perkins, Ad hoc Networks, Addison Wesley.
5. Judith Hurwitz, Robin Bllor, Marcia Kaufmann, Fern Halper, Cloud Computing forDummies, 2009.

| **CSE-308** | **Web Engineering** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3** |
| **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** | | | | | | | |
| **CO1** | Learn the basic concepts of information and web architecture. | | | | | | |
| **CO2** | Learn about the skills that will enable to design and build high level web enabled applications. | | | | | | |
| **CO3** | Understand the applicability of Java Script as per current software industry standards. | | | | | | |
| **CO4** | Acquaint the latest programming language for the implementation of object based and procedure based applications using Python. | | | | | | |

### 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

**Java Script**: Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Primitives, Operations, and Expressions, Screen Output and Keyboard Input, Control Statements, Object Creation and Modification, Arrays, Functions, Constructors, Pattern Matching Using Regular Expressions, Errors in Scripts

### 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**

1. By Peter Morville, Louis Rosenfeld, “Information Architecture on the World Wide Web”, O'Reilly Media, 2006.
2. Robert W. Sebesta, “Programming The World Wide Web”, Eight Edition, Pearson India, 2015.
3. Kenneth A. Lambert, “The Fundamentals of Python: First Programs”, 2011, Cengage Learning.

**Reference Book**

1. Thomas A Powell, “HTML The Complete Reference”, Tata McGraw Hill Publications.

| CSE-310 | Software Engineering | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **1** | **-** | **3.5** | **75** | **25** | **100** | **3** |
| **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

**Introduction**: Introduction to Software Engineering, Software Characteristics, Software Crisis, The Evolving role of Software, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, RAD, V Model.

## Unit-II

**Software Requirement Specification** :Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Data Flow Diagrams, Decision Tables, SRS Document, IEEE Standard for SRS.

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

## Unit-III

**Software Design**: Basic Concept of Software Design, Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion.

**Design Strategies:** Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design.

**Software Measurement and Metrics:** Various Size Oriented Measures: Halstead‟s Software Science, Function Point (FP) Based Measures, COCOMO, Cyclomatic Complexity Measures: Control Flow Graphs.

## Unit-IV

**Software Construction:**Software construction fundamentals, minimizing complexity, Top-Down and Bottom –Up programming, structured programming, Compliance with Design and Coding Standards.

**Testing:** Testing Objectives, Unit Testing, Integration Testing, system testing, Acceptance Testing, Regression Testing, Structural Testing, FunctionalTesting, debugging.

**Maintenance**: key issues, Types of software Maintenance, Cost of Maintenance, Software Re-Engineering.

**Text Books:**

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.

**Reference Books:**

1. Pankaj Jalote, Software Engineering, Wiley India.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
3. Ian Sommerville, Software Engineering, Addison Wesley.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **HS-303** | **Business Intelligence & Entrepreneurship** | | | | | | |
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **-** | **-** | **4** | **75** | **25** | **100** | **3** |

| **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:**

1. Small-Scale Industries and Entrepreneurship. Himalaya Publishing House, Delhi -Desai, Vasant, 2003.
2. Entrepreneurship Management -Cynthia, Kaulgud, Aruna, Vikas Publishing House, Delhi, 2003.
3. Entrepreneurship Ideas in Action- L. Greene, Thomson Asia Pvt. Ltd., Singapore, 2004.

| **CSE-312** | **Web Engineering Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **3** | **1.5** | **40** | **60** | **100** | **3 Hrs.** |
| **Purpose** | To introduce the concepts of HTML5, JavaScript and Python. | | | | | | |
| **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 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.

[6] Incorporate a quest book into the Diary Food Webpage and use Java Script to build validations into the form.

[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

[10] Write program in Java Script for pattern matching using regular expressions and errors in scripts.

[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.

[12] Write the Python functions for linear search, binary search, selection sort, Bubble Sort, Insertion Sort and converting Fibonacci to a linear algorithm.

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

| CSE-314 | Essentials of Information Technology Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **3** | **1.5** | **40** | **60** | **100** | **3 Hrs.** |
| **Purpose** | To introduce the concepts of Object Oriented Programming using Java and RDBMS | | | | | | |
| **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

| **CSE-316** | **Software Engineering Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **-** | **-** | **3** | **1.5** | **40** | **60** | **100** | **3** |
| **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 Science & Engineering)
## Credit Based Scheme of Studies/Examination
## Semester VII

| S. No. | Course No. | Subject | L:T:P | Hours/ Week | Credits | Examination Schedule | | | | Duration of Exam (Hrs.) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | **Major Test** | **Minor Test** | **Practical** | **Total** | |
| 1 | CSE 401 | Unix & Linux Program-ming | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 2 | CSE 403 | Computer Graphics and Animation | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 3 | PE-I | Elective* – I | 3:0:0 | 3 | 3.0 | 75 | 25 | 0 | 100 | 3 |
| 4 | PE-II | Elective* – II | 3:0:0 | 3 | 3.0 | 75 | 25 | 0 | 100 | 3 |
| 5 | CSE 405 | Computer Graphics Lab | 0:0:2 | 2 | 1.0 | 0 | 40 | 60 | 100 | 3 |
| 6 | CSE 407 | Project-I** | 0:0:8 | 8 | 4.0 | 0 | 100 | 100 | 200 | 3 |
| 7 | CSE 409 | Unix & Linux Program-ming Lab | 0:0:2 | 2 | 1.0 | 0 | 40 | 60 | 100 | 3 |
| 8 | CSE 411 | Seminar | 0:0:2 | 2 | 1.0 | 0 | 100 | 0 | 100 | |
| 9 | CSE 413 | Industrial Training (Viva-Voce)*** | | | | | 100 | 0 | 100 | |
| | | **Total** | | **28** | **21.0** | **300** | **480** | **220** | **1000** | |

| Code | PE-I | Code | PE-II |
|---|---|---|---|
| CSE-415 | Object Oriented Software Engineering | CSE-421 | Agile Software Engineering |
| CSE-417 | Cyber Security | CSE-423 | Big Data and Analytics |
| CSE-419 | Cryptography & Information Security | CSE-425 | Expert Systems |

**Note:**

***The students will choose any two departmental electives courses out of the given elective list in 7thSemester.**

****Project should be initiated in 7thsemester beginning, and should be completed by the end of 8thsemester with good Report and power-point Presentation etc.**

*****4-6 weeks hand on training to be done after 7thSemester Exams.**

| **CSE-401** | **Unix & Linux Programming** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **Introduces commands and numerous programming concepts and application domains to cover important topics for implementation of the Unix programming concepts.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To learn basic and advanced Unix Commands. | | | | | | |
| **CO2** | Expose the role of filters and file compression techniques. | | | | | | |
| **CO3** | To explore knowledge of programming language development tools. | | | | | | |
| **CO4** | To expand knowledge of Unix/Linux system administration and networking. | | | | | | |

### Unit I: Basic Command Usage

Linux Startup: User accounts, accessing Linux - starting and shutting processes, Logging in andLogging out, Unix commands like zip, unzip, pack, unpack, compress, uncompress, Shell Programming, Unix file system: Linux/Unix files, i-nodes and structure, file systemrelated commands, Shell as command processor, shell variables, creating command substitution, scripts, functions, conditionals, loops, customizing environment

### Unit II: Filters and File Compression

Regular Expressions and Filters: Introducing regular expressions patterns, syntax, character classes, quantifiers, introduction to grep, egrep, sed, programming with awk and perl, File Compression Techniques: data redundancy elimination using fingerprint generation deduplication and data similarities removal using delta techniques for data reduction storage, parallel compression with Xdelta utility.

### Unit III: Program Development Tools

The C Environment: C compiler, vi editor, compiler options, managing projects, memorymanagement, use of makefile, cmake, dependency calculations, memory management – static and dynamic memory, static and dynamic libraries, dynamic loader, debugging tools like gdb, fixed-size and variable-size blocks of data files chunks divisor chunking techniques like Frequency Based Chunking and Content Defined Chunking Unix based open source coding.

### Unit IV: Process Control

Processes in Linux: Processes, starting and stopping processes, initialization processes, rc andinit files, job control - at, batch, cron, time, network files, security, privileges, authentication,password administration, archiving, Signals and signal handlers, Threading, Linux I/O system, Networking tools like ping, telnet, ftp, route, Firewalls, Backup and Restore tar, cpio, dd.
Case Study: PCOMPRESS open source free software

**Text Books**:

1. John Goerzen: Linux Programming Bible, IDG Books, New Delhi, 2014.
2. Sumitabha Das: Unix – Concept and Applications, Fourth Edition TMH, 2015.
3. Neil Matthew, Richard Stones: Beginning Linux Programming, 4th. Edition, Wrox-Shroff, 2011.
4. Welsh & Kaufmann: Running Linux, O'Reiley& Associates, 2013.

**Reference Book:**

1. B.M. Harwani, *Unix and Shell Programming*, Oxford University Press, 2013.

| CSE-403 | Computer Graphics and Animation | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **03 Hrs.** |
| **Purpose** | **Introduces Computer Graphics that help in designing different kinds of static and movable objects.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Explore the background and standard line and circle drawing algorithms. | | | | | | |
| **CO2** | Exposure of various transformation approaches and its comparative analysis. | | | | | | |
| **CO3** | Illustrate Projection and clipping with explore different techniques. | | | | | | |
| **CO4** | Apply design principles to create different curves and explore hidden lines and surface techniques. | | | | | | |

## Unit 1

Computer Graphics applications, Display Devices, Point & Positioning Devices, Plotting Techniques for point and Line, Line drawing algorithms: DDA, Bresenhams's Circle drawing algorithms, Filled area algorithms: Scan line, Polygon filling algorithms, Boundary filled algorithms.

## Unit-2

Window to view port transformation, Window to view port mapping, Two Dimensional transformation: translation, scaling, rotation, reflection and Shear, Homogeneous Coordinate system.

3-D transformation: Rotation, Shear, translation, Numerical Problems of transformation viewing pipeline.

## Unit-3

Clipping: Point & Line clipping algorithm, 4-bit code algorithm, Cohen-Sutherland Line clipping algorithms, Liang-Barsky line clipping algorithms. Polygon clipping: Sutherland-Hodgeman Polygon clipping algorithm. Curve clipping, Text clipping.

Projection: Parallel, Perspective, Vanishing Points.

## Unit-4

Representation of 3-D Curves and Surfaces: interpolation and approximation alpines, parametric conditions, Geometric continuity conditions, Beizer curves and surfaces: properties of beizer curves, beizer surfaces.

Hidden Surfaces removal: Hidden surface elimination, depth buffer algorithm, scan line coherence and area coherence algorithm, priority algorithm

**Text Books**

1. Donald Hearn &M.Pauline Baker, Computer Graphics, 2nd Edition, Pearson Education.
2. William M. Newmann& Robert F. Sproull, Principles ofInteractive Computer Graphics, Tata McGraw-Hill Second Edition, New Delhi, India.
3. Zhigang Xiang & Roy A Plastock , Computer Graphics, Second Edition, Schaum's Outline, Tata McGraw Hill Education Private Limited, New Delhi, India.

**References Book**

1. Foley, van Dam, Feiner, and Hughes. Computer Graphics: Principles and Practice, 3rd edition in C.

| CSE-415 | Object Oriented Software Engineering | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
| 3 | 0 | 0 | 3 | 75 | 25 | 100 | 3 Hrs. |
| Purpose | **To provide the thorough knowledge to use the concepts and their design attributes for Object Oriented Software Engineering approaches and platforms to solve real time problems.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| CO1 | To learn the basic concepts of object oriented systems and software engineering. | | | | | | |
| CO2 | To get exposure of various object modeling methodologies, tools for analyzing and designing software based systems using UML. | | | | | | |
| CO3 | To explore problems using Use Cases, analyzing relations, responsibilities and collaborations among classes and their behavior in problem domain. | | | | | | |
| CO4 | To evaluate object oriented design processes using models, design patterns, interfaces designs and communication mechanisms for performing required tasks. | | | | | | |

### Unit - I

An Overview of Object-Oriented system Development, Objects Basis, 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 - II

Object Oriented Methodologies:Rumbaugh Methodology, Jacobson Methodology, BoochMethodology, Patterns, Frameworks, The Unified approach, Unified Modeling Language (UML)

### Unit - III

Object Oriented Analysis Process, Use Case Driven Object Oriented Analysis, Use Case Model,Object Analysis: Classification, Classification Theory, Approaches for identifying classes, Responsibilities and Collaborators, Identifying Object Relationships, Attributes and Methods: Associations, Super-Sub Class relationships, A-Part-of-Relationships-Aggregation, Class Responsibilities, Object Responsibilities.

### Unit - IV

Object Oriented Design process and Design Axioms, Corollaries, Design Patterns, Designing Classes: Object Oriented Design Philosophy, UML Object Constraint Language, Designing Classes: The Process, Class Visibility, Refining Attributes, Designing Methods and Protocols, Packages and Managing classes, View Layer: Designing Interface objects,Designing View layer Classes, Macro and Micro Level Interface Design Process.

**Text Books:**

1. Ali Bahrami, Object Oriented Systems Development, McGraw HillPublishing Company Limited, New Delhi, 2013.
2. Rumbaugh *et al.*, Object Oriented Modeling and Design, PHI, 2006.
3. Robert Laganière and Timothy C. Lethbridge, Object-Oriented Software Engineering: Practical Software Development, McGraw-Hill Publishing Company Limited, New Delhi, Sixth Print 2008.

**Reference Books:**

1. Ivar Jacobson, MagnosChristerson, Patrick Jonsson, Gunnar Overgaard, Object-oriented Software Engineering: A Use Case Driven Approach, Pearson Education, New Delhi, Seventh Edition Reprint, 2009.
2. David C. Kung, Object-Oriented Software Engineering: An Agile Unified Methodology, McGraw-Hill Publishing Company Limited, New Delhi, 2013.
3. Bernd Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International, Third Edition, 2013.

| CSE-417 | Cyber Security | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **-** | **-** | **3** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | To gain a broad understanding in order to get predictive ways out related to cyber security. | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | To facilitate the basic knowledge of cyber security. | | | | | | |
| **CO2** | To explore and sort issues related to different types of activities in cyber crime. | | | | | | |
| **CO3** | To get enable to fix the various cyber attacks. | | | | | | |
| **CO4** | To deal with the digital forensics and related scenarios of cyber crimes. | | | | | | |

## Unit I

**Introduction:** Introduction and Overview of Cyber Crime, Nature and Scope of Cyber Crime, Types of Cyber Crime: crime against individual,Crime against property, Cyber extortion, Drug trafficking, cyber terrorism.

Need for Information security, Threats to Information Systems, Information Assurance, Cyber Security, and Security Risk Analysis.

## Unit 2

**Cyber Crime Issues**: Unauthorized Access to Computers, Computer Intrusions, Viruses and Malicious Code, Internet Hacking and Cracking, Virus and worms, Software Piracy, Intellectual Property, Mail Bombs, Exploitation, Stalking and Obscenity in Internet, Password Cracking, Steganography, Key loggers and Spyware, Trojan and backdoors, phishing, DOS and DDOS attack, SQL injection, Buffer Overflow.

## Unit 3

**Introduction to cyber attacks:**passive attacks, active attacks.

Cyber crime prevention methods, Application security (Database, E-mail and Internet), Data Security Considerations-Backups, Archival Storage and Disposal of Data, Security Technology-Firewall and VPNs, Intrusion Detection, Access Control,Hardware protection mechanisms,OS Security

## Unit 4

**Digital Forensics:** Introduction to Digital Forensics, historical background of digital forensics, Forensic Software and Hardware, need for computer forensics science, special tools and techniques digital forensic life cycle, challenges in digital forensic.

**Law Perspective**: Introduction to the Legal Perspectives of Cybercrimes and Cyber security, Cybercrime and the Legal Landscape around the World, Why Do We Need Cyber laws, The Indian IT Act, Cybercrime Scenario in India, Digital Signatures and the Indian IT Act, Cybercrime and Punishment.

**Text Books:**

1. Nelson Phillips and EnfingerSteuart, “Computer Forensics and Investigations”, Cengage Learning, New Delhi, 2009.

**Reference Books:**

1. Robert M Slade,” Software Forensics”, Tata McGraw - Hill, New Delhi, 2005.
2. SunitBelapure and Nina Godbole, “Cyber Security: Understanding Cyber Crimes, Computer Forensics And Legal Perspectives”, Wiley India Pvt. Ltd.

| CSE-419 | Cryptography and Information Security | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **0** | **0** | **3** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | The course will be useful for students who plan to do research/product development/analysis in areas related to secure computing in their career. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| CO1 | To learn basics of network security and cryptography. | | | | | | |
| CO2 | Exposing the knowledge about network authentication mechanism, with security algorithms. | | | | | | |
| CO3 | To explore the knowledge of key exchange protocols. | | | | | | |
| CO4 | To realize the effect on digitized security. | | | | | | |

## Unit 1: Basics of Cryptography

Introduction to cryptography, security threats, types of cryptography, Classical cryptography and their cryptanalysis, perfect secrecy, Shannon's theorem, stream ciphers, Security attacks

## Unit 2: Authentication Mechanism and Security Algorithms

**A**ccess control mechanism, Discretionary v/s mandatory access control, CPA-secure encryption, Pseudorandom permutations, practical block ciphers (3-DES, AES), RSA, modes of operation, MACs, Hash functions-Tiger Hash, Gear hash, pseudorandom generators , Public key infrastructure.

## Unit3: Key Exchange Protocols

CCA-secure encryption, Diffie-Hellman key exchange, Public key crypto systems (El Gamal, Paillier, Rabin, Goldwasser-Micali), Key exchange protocols, example protocol such as PGP, Kerberos, IPSEC/VPN, SSL, S/MIME etc., PKCSv1.5.

## Unit 4: Digitized Security

Digital signatures,-MD5, SHA1, Rabin Finger Print, digital certificates, DSS, firewall and intrusion detection systems, Byzantine agreement, secure multiparty computation, interactive proof systems

**Text Books**:

1. Y. Lindell and J. Katz. Introduction to Modern Cryptography. MIT press, 2012.
2. OedGoldreich. Foundations of Modern cryptography: Parts I and II, Cambridge Press, 2011.
3. A. Menezes, P.C. Van Oorschot and S.A. Vanstone. Handbook of Applied Cryptography, CRC Press, 2010.
4. William Stalling, Cryptography and Network Security: Pearson Education, 2013.

**Reference Books:**

1. Michael EWhitman& Herbert J. Mattord, Principles of Information Security, Vikash Publishing House PVT. LTD., New Delhi, 2015.
2. Charles P. Pfleeger, Security in Computing, 4th Edition, Prentice Hall, 2011.
3. Jeff Crume, Inside Internet Security Addison Wesley, 2014.

| **CSE-421** | **Agile Software Engineering** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **0** | **0** | **3** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **Introduces the business value of adopting Agile approaches and provide complete understanding of the Agile development practices** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Understand the background and driving forces for taking an Agile approach to software development. | | | | | | |
| **CO2** | Understand the business value of adopting Agile approaches. | | | | | | |
| **CO3** | Drive development with unit tests using Test Driven Development. | | | | | | |
| **CO4** | Apply design principles and refactoring to achieve Agility. | | | | | | |

## Unit I: Fundamentals of Agile

The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools

## Unit II: Agile Scrum Framework

Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management.

## Unit III: Agile Testing

The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester.

## Unit IV: Agile Software Design and Development

Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control.

**Text Books**:

1. Ken Schawber, Mike Beedle, *Agile Software Development with Scrum*, Pearson publications.
2. Robert C. Martin, *Agile Software Development, Principles, Patterns and Practices*, Prentice Hall.
3. Lisa Crispin, Janet Gregory, *Agile Testing: A Practical Guide for Testers and Agile Teams*, Addison Wesley.

**Reference Books:**

1. Alistair Cockburn, *Agile Software Development: The Cooperative Game*, Addison Wesley.
2. Mike Cohn, *User Stories Applied: For Agile Software,* Addison Wesley.

| CSE-423 | Big Data and Analytics | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
| 3 | 0 | 0 | 3 | 75 | 25 | 100 | 3 Hrs. |
| Purpose | To provide knowledge of Big Data Analytics and Distributed File Systems. | | | | | | |
| Course Outcomes (CO) | | | | | | | |
| CO1 | To learn in details the concepts of big data. | | | | | | |
| CO2 | Expose the criteria of big data analytics and big data storage. | | | | | | |
| CO3 | To explore knowledge of big data compression techniques. | | | | | | |
| CO4 | To explore learning of big data tools and state-of-the-art knowledge with implementation for big data. | | | | | | |

### Unit I: Big Data Background

Big data definition and features of big data, big data value, development of big data, challenges of big data, NoSQL databases, technologies related to big data including cloud computing, Internet of Things, data center, Hadoop, relationship between IoT and big data, relationship between hadoop and big data, big data generation and acquisition includes data collection, data transmission, data pre-processing, big data applications.

### Unit II: Big Data Analytics and Storage

Big data analysis, big data analytic methods and tools, Pig, Hive, Flume, Mahout, Big data storage, distributed storage system for massive data, storage mechanism for big data GFS, HDFS, HBase, MongoDB, Cassandra, big data storage deduplication techniques, fixed-size and variable-size blocks based deduplication, content defined chunking, frequency based chunking, byte and multi-byte indexing techniques, Cloud storage.

### Unit III: Big Data Compression

Big data delta compression, Xdelta implementation, Message Digest (MD5), Secure Hash Algorithm (SHA-1/SHA-256), Gear Hash, Tiger Hash, Rabin and Incremental Secure Fingerprint based deduplication, lossless duplicate and similar data elimination approaches, Parallel deduplication and compression using PCOMPRESS, Scalable Decentralized Deduplication Store (SDDS) using Cassandra.

### Unit IV: Big Data Processing

Installation procedure with system requirements for Apache Hadoop, Cassandra, Spark, Pig, Hive, HBase, MongoDB large scale distributed storage systems, Map Reduce programming model working, YARN architecture, Apache Pig and Hive architecture, Single node and Multi-nodes Hadoop Cluster Set up and running a Big Data example, NoSQL implementation.

**Text Books**:

1. "Big Data" by Viktor Mayer-Schönberger, Kenneth Cukier, ISBN:978-0544002692, Eamon Dolan/Houghton Mifflin Harcourt 2013.
2. “Big Data Now”, by O’Reilly Media Inc., ASIN: B0097E4EBQ, O’Reilly 2012.
3. “Hadoop Operation”, by Eric Sammer, ISBN: 978-1449327057, O’Reilly 2012.
4. “MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems”, by Donald Miner, Adam Shook, ISBN:978-1449327170, O’Reilly 2012.

**Reference Books:**

1. “Programming Hive”, by Edward Capriolo, ISBN: 978-1449319335,O’Reilly 2012.
2. “HBase: the Definitive Guide”, by Lars George, ISBN: 978-1449396107, O’Reilly 2011.
3. “Mahout in Action”, by Sean Owen, Robin Anil, Ted Dunning, Ellen Friedman, ISBN: 978-1935182689, Manning 2011.
4. “Programming Pig”, by Alan Gates, ISBN: 978-1449302641, O’Reilly 2011.
5. “Cassandra, the Definitive Guide”, by Eben Hewitt ISBN: 978-1449390419 O’Reilly 2011.
6. “MongoDB: The Definitive Guide” by Kristina Chodorow, Michael Dirolf,ISBN: 978-1449381561, O’Reilly, 2010.

| CSE-425 | Expert Systems |
|---|---|

| Lecture | Tutorial | Practical | Credit | Major Test | Minor Test | Total | Time |
|---|---|---|---|---|---|---|---|
| 3 | 0 | 0 | 3 | 75 | 25 | 100 | 3 Hrs. |
| **Purpose** | In this course the student will learn the methodologies used to transfer the knowledge of a human expert into an intelligent program that can be used to solve real-time problems. | | | | | | |
| **Course Outcomes(CO)** | | | | | | | |
| **CO1** | Examining the fundamentals and terminologies of expert system. | | | | | | |
| **CO2** | To facilitate students to implement various knowledge representation techniques for acquisition and validate various structures in experts system domain. | | | | | | |
| **CO3** | Signifying AI techniques to solve social, industrial and environmental problems. | | | | | | |
| **CO4** | Application of professional aspects in multi-disciplinary approach to meet global standards towards design, realizing and manufacturing. | | | | | | |

## Unit-1

Introduction to AI programming languages, Blind search strategies, Breadth first – Depth first – Heuristic search techniques Hill Climbing – Best first – A Algorithms AO* algorithm – game tress, Min-max algorithms, game playing – Alpha beta pruning.

Knowledge representation issues predicate logic – logic programming Semantic nets- frames and inheritance, constraint propagation; Representing Knowledge using rules, Rules based deduction systems.

## Unit-2

Introduction to Expert Systems, Architecture of expert system, Representation and organization of knowledge, Basics characteristics, and types of problems handled by expert systems.

**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.

### TEXT BOOKS

1. Elain Rich and Kevin Knight, “Artificial Intelligence”, Tata McGraw-Hill, New Delhi, 2008.
2. Waterman D.A., “A Guide to Expert Systems”, Addison Wesley Longman, 1985.

### REFERENCE BOOKS

1. Staurt Russel and other Peter Norvig, “Artificial Intelligence – A Modern Approach”, Prentice Hall, 1995.
2. Patrick Henry Winston, “Artificial Intelligence”, Addison Wesley, 1979.
3. Patterson, Artificial Intelligence & Expert System, Prentice Hall India,1999.
4. Hayes-Roth, Lenat and Waterman: Building Expert Systems, Addison Wesley, 1983.
5. Weiss S.M. and Kulikowski C.A., “A Practical Guide to Designing Expert Systems”, Rowman &Allanheld, New Jersey, 2011.

| **CSE-405** | **Computer Graphics Lab** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **2** | **1** | **40** | **60** | **100** | **3Hrs.** |
| **Purpose** | **To Design and implement various Line and Circle Drawing Algorithms.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To Implement basic algorithms related to Line & Circle Drawing. | | | | | | |
| **CO2** | Implement various Line & Circle Drawing Algorithms. | | | | | | |
| **CO3** | Hands on experiments on 2-D transformations. | | | | | | |
| **CO4** | Conceptual implementation of Clipping and other drawing algorithms.. | | | | | | |

**List of Practicals**

1. Write a program to implement DDA line drawing algorithm.
2. Write a program to implement Bresenham’s line drawing algorithm.
3. Implement the Bresenham’s circle drawing algorithm.
4. Write a program to draw a decagon whose all vertices are connected with every other vertex using lines.
5. Write a program to move an object using the concepts of 2-D transformations.
6. Write a program to implement the midpoint circle drawing algorithmany Object Oriented Programming Languagelike Python, C++,Java.
7. Implement the line clipping algorithm using any Object Oriented Programming Language like Python, C++, Java.
8. Implement boundary fill algorithm using any Object Oriented Programming Language like Python, C++, Java.
9. Implement the depth buffer algorithm using any Object oriented language like Python, C++,Java.
10. Perform the Polygon Clipping Algorithm using any Object oriented language like Python, C++,Java.
11. Draw a Rectangle using Bresenham’s and DDA Algorithm using any Object oriented language like Python, C++,Java.

| CSE-409 | Unix & Linux ProgrammingLab | | | | | | |
|---|---|---|---|---|---|---|---|
| Lecture | Tutorial | Practical | Credit | Minor Test | Practical | Total | Time |
| 0 | 0 | 2 | 1 | 40 | 60 | 100 | 3 Hrs. |
| Purpose | To provide experimental knowledge of Unix & Linux Programs | | | | | | |
| Course Outcomes (CO) | | | | | | | |
| CO | Exploring knowledge by implementation of programs using UNIX/LINUX. | | | | | | |

**List of Practicals**

1. Familiarize with Unix/Linux logging/logout and simple commands.
2. Familiarize with vi editor.
3. Using Bash shell develop simple shell programs.
4. Develop advanced shell programs using grep, fgrep&egrep.
5. Compile and debug various C programs using different options.
   Content defined chunking, frequency based chunking, delta/Xdelta, Rabin Fingerprint Generator, Parallel Compression pcompress.
6. Learning of installation and upgradation of Linux operating system.
7. Install Linux on a PC having some other previously installed operating system. All OSs should be usable.
8. As supervisor create and maintain user accounts, learn package installation, taking backups, creation of scripts for file and user management, creation of startup and shutdown scripts using at, cron etc.

**Note: At least 5 to 10 more exercises are to be given by the teacher concerned.**

## Bachelor of Technology (Computer Science & Engineering)

## Credit Based Scheme of Studies/Examination
## Semester VIII

| S. No. | Course No. | Subject | L:T:P | Hours/ Week | Credits | Examination Schedule | | | | Duration of Exam (Hrs.) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | **Major Test** | **Minor Test** | **Practical** | **Total** | |
| 1 | CSE 402 | Neural Networks & Fuzzy Logic | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 2 | PE-III | Elective*-III | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 3 | PE-IV | Elective* - IV | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 4 | CSE 404 | Mobile Apps Development | 4:0:0 | 4 | 4.0 | 75 | 25 | 0 | 100 | 3 |
| 5 | CSE 406 | Mobile Apps Development Lab | 0:0:2 | 2 | 1.0 | 0 | 40 | 60 | 100 | 3 |
| 6 | CSE 408 | Computer Hardware & Troubleshooting Lab | 0:0:2 | 2 | 1.0 | 0 | 40 | 60 | 100 | 3 |
| 7 | CSE 410 | Project-II | 0:0:16 | 16 | 8.0 | 0 | 100 | 100 | 200 | 3 |
| | | **Total** | | **36** | **26.0** | **300** | **280** | **220** | **800** | |
| 8 | CSE 424 | General Fitness & Professional Aptitude** | | | | | 100 | | 100 | 8 |

| Code | PE-III | Code | PE-IV |
|---|---|---|---|
| CSE-412 | Software Testing | CSE-418 | Parallel Computing |
| CSE-414 | Graph Theory | CSE-420 | Cloud Computing |
| CSE-416 | Data Mining | CSE-422 | Natural Language Processing |

**Note:**

***The students will choose any two departmental electives courses out of the given elective list in VIII Semester.**

****CSE 424 will be non credit subject and only grade will be awarded to student based on his/her performance in the examination.**

| CSE-402 | Neural Networks & Fuzzy Logic | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **To provide knowledge of various artificial neural networks, fuzzy logic techniques and Genetic Engineering approach for optimization** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To learn the basics of artificial neural networks concepts. | | | | | | |
| **CO2** | Expose detailed explanation of various neural networks architecture. | | | | | | |
| **CO3** | To explore knowledge of special types of Artificial neural networks. | | | | | | |
| **CO4** | To explore fuzzy logic techniques and genetic algorithms in neural networks. | | | | | | |

## Unit I: Fundamentals of Artificial Neural Networks

Introduction: Concepts of neural networks, Characteristics of Neural Networks, 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, perceptron learning and training, Classification, linear Separability

## Unit II: Neural Networks

Hopfield nets: Structure, training, and applications, Back Propagation: Concept, Applications and Back Propagation Training Algorithms. Counter Propagation Networks: Kohonan Network, Grossberg Layer & Training, applications of counter propagation, Image classification.

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

## Unit III: Special Neural Networks

ART: ART architecture, ART classification operation, ART implementation and characteristicsof ART. Image Compression Using ART, Optical Neural Networks: Vector Matrix Multipliers, Hop field net using Electro optical matrix multipliers, Holographic correlator, Optical Hopfield net using Volume Holograms, Cognitrons and Neocognitrons: structure and training.

## Unit IV: Fuzzy Logic

Fuzzy Logic:Introduction to Fuzzy Logic, Classical and Fuzzy Sets: Overview of Classical Sets,Membership Function, Fuzzy rule generation, Operations on Fuzzy Sets: Compliment, Intersections,Unions, Combinations of Operations, Aggregation Operations, Fuzzy Arithmetic: Fuzzy Numbers,Linguistic Variables, Arithmetic Operations on Intervals & Numbers, Lattice of Fuzzy Numbers, Fuzzy Equations, Introduction of Neuro-Fuzzy Systems, Architecture of Neuro Fuzzy Networks, Genetic Algorithms: genetic algorithm implementation in problem solving and working of genetic algorithms evolving neural networks, Differential Evolution optimization for engineering problems.

**Text Books**:

1. Li Min Fu, “Neural Networks in Computer Intelligence”, McGraw-Hill, Inc. 2012.
2. S N Sivanandam, “Neural Networks using MATLAB 6.0”, TMH, 4th. Reprint 2015.
3. S N Sivanandam, “Principles of Soft Computing”, 2nd. Edition, Wiley, Reprint 2014.

**Reference Books:**

1. Simon Haykin, “Neural Networks: A Comprehensive Foundations”, Prentice-Hall International, New Jersey, 2013.
2. Freeman J.A. & D.M. Skapura, “Neural Networks: Algorithms, Applications and Programming Techniques”, Addison Wesley, Reading, Mass, 2014.

| CSE-412 | Software Testing | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | To provide an understanding of concepts and techniques for testing software and assuring its quality. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Expose the criteria and parameters for the generation of test cases. | | | | | | |
| **CO2** | Learn the design of test cases and generating test cases. | | | | | | |
| **CO3** | Be familiar with test management and software testing activities. | | | | | | |
| **CO4** | Be exposed to the significance of software testing in web and Object orient techniques. | | | | | | |

## UNIT – I

**Introduction:** Overview of software evolution, SDLC, Testing Process, Terminologies in Testing: Error, Fault, Failure, Verification, Validation, Difference between Verification and Validation, What is software testing and why it is so hard? Test Cases, Test Oracles,Testing Process, Limitations of Testing.

## UNIT - II

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

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

## UNIT - III

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

**Testing Activities:** Unit Testing, Levels of Testing, Integration Testing, System Testing, Debugging, Domain Testing.

## UNIT - IV

**Object oriented Testing**: Definition, Issues, Class Testing, Object Oriented Integration and System Testing.

**Testing Web Applications**: What is Web testing?, User interface Testing, Usability Testing, Security Testing, Performance Testing, Database testing, Post Deployment Testing.

**TEXT BOOKS:**

1. Naresh Chauhan “Software Testing Principles and Practices” Oxford Publications, 2012.
2. Louise Tamres, “Software Testing”, Pearson Education Asia, 2002.
3. Robert V. Binder, “Testing Object-Oriented Systems-Models, Patterns and Tools”, Addison Wesley, 1999.
4. William Perry, “Effective Methods for Software Testing”, John Wiley & Sons, New York, 1995.

**REFERENCE BOOKS:**

1. CemKaner, Jack Falk, Nguyen Quoc, “Testing Computer Software”, Second Edition, Van Nostrand Reinhold, New York, 1993.
2. K.K. Aggarwal & Yogesh Singh, “Software Engineering”, 2$^{nd}$ Ed., New Age International Publishers, New Delhi, 2005.
3. Boris Beizer, “Software Testing Techniques”, Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.
4. Boris Beizer, “Black-Box Testing – Techniques for Functional Testing of Software and Systems”, John Wiley & Sons Inc., New York, 1995.
5. Gopalaswamy Ramesh, Srinivasan Desikan, Software Testing : Principles and Practices, Pearson India, 2005.

| **CSE-414** | **Graph Theory** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3Hrs.** |
| **Purpose** | **To familiarize the students with the fundamentals of Graph Theory and Graph algorithms.** | | | | | | |
| **Course Outcomes** | | | | | | | |
| **CO1** | To get enabled about the various concepts of graph theory. | | | | | | |
| **CO2** | To explore different trees, graphs and algorithms. | | | | | | |
| **CO3** | To deal with the concept of planar graph and its related algorithms. | | | | | | |
| **CO4** | To implement the concept of vectors, colouring, covering and partitioning of a graph. | | | | | | |

## UNIT- I

**Introduction :** Graphs, Isomorphism, Sub graphs, some basic properties, various example of graphs & their sub graphs, walks, path & circuits, connected graphs, disconnected graphs and component, directed graphs, types of directed graphs,Euler graphs, various operation on graphs, Hamiltonian paths and circuits, the traveling sales man problem.

## UNIT- II

**Trees:** Trees and fundamental circuits, distance diameters, radius and pendent vertices, rooted and binary trees, on counting trees, spanning trees, fundamental circuits, finding all spanning trees of a graph and a weighted graph, algorithms of primes, Kruskal and Dijkstra Algorithms.

## UNIT- III

**Fundamentals of Cut sets:** Cut sets Cuts sets and cut vertices, some properties, all cut sets in a graph, fundamental circuits and cut sets, connectivity and separability, network flows.

**Planar Graphs:** Planer graphs, different representation of a planar graph, combinatorial and geometric dual: Kuratowski graphs, detection of planarity, geometric dual, Discussion on criterion of planarity, thickness and crossings.

## UNIT- IV

**Vector:** Vector space of a graph and vectors, basis vector, cut set vector, circuit vector, circuit and cut set subspaces, Matrix representation of graph – Basic concepts; Incidence matrix, Circuit matrix, Path matrix, Cut-set matrix and Adjacency matrix.

**Graph Colouring, covering and partitioning:**Colouring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, matching, covering, four color problem.

**Text Books:**

1. Deo, N, Graph theory with applications to Engineering and Computer Science, PHI.
2. Gary Chartrand and Ping Zhang, Introduction to Graph Theory, TMH.

**Reference Books:**

1. Robin J. Wilson, Introduction to Graph Theory, Pearson Education.
2. Harary, F, Graph Theory, Narosa Publication.
3. Bondy and Murthy: Graph theory and application. Addison Wesley.
4. V. Balakrishnan, Schaum's Outline of Graph Theory, TMH.
5. GeirAgnarsson, Graph Theory: Modeling, Applications and Algorithms, Pearson Education.

| **CSE-416** | **Data Mining** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **To provide the knowledge of data mining and its techniques.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To learn data mining concepts in details. | | | | | | |
| **CO2** | Expose the criteria for data generalization. | | | | | | |
| **CO3** | To explore knowledge of mining associations, correlations and classification. | | | | | | |
| **CO4** | To evaluate various types of data mining. | | | | | | |

### Unit I: Data Mining and Data Preprocessing

Introduction :Data Mining, Functionalities, Data Mining Systems classification, Integration with Data Warehouse System, Data summarization, data cleaning, data integration and transformation, data reduction. Data Warehouse:Need for Data Warehousing, Paradigm Shift, Business Problem Definition, Operational and Information Data Stores, Data Warehouse Definition and Characteristics, Data Warehouse Architecture and Implementation, OLAP.

### Unit II: Data Generalization

Data Mining Primitives, Query Language and System Architecture, Concept Description, Data generalization, Analysis of attribute relevance, Mining descriptive statistical measures in large databases, Data deduplication methodologies.

### Unit III: Mining Associations and Correlations

Mining association rules in large databases:Association rule mining, Mining single dimensional boolean association rules from transactional databases, mining multilevel association rules from transaction databases, Relational databases and data warehouses, correlation analysis, classification and prediction, Data redundancy detection and elimination techniques.

### Unit IV: Cluster Analysis and Mining

Introduction to cluster analysis, Mining complex type of data: Multidimensional analysis and descriptive mining of complex data objects, Spatial databases, Multimedia databases, Mining timeseries and sequence data, Mining text databases, Mining World Wide Web, Data Chunking Techniques.

**Text Books**

1. J.Han, M.Kamber, Data Mining: Concepts and Techniques, Academic Press, Morgan Kanfman Publishers, 2015.
2. Pieter Adrians, DolfZantinge, Data Mining, Addison Wesley 2013.
3. C.S.R. Prabhu, Data Ware housing: Concepts, Techniques, Products and Applications, Prentice Hall of India, 2014.

**Reference Books**

1. Berry and Lin off, Mastering Data Mining: The Art and Science of Customer Relationship Management, John Wiley and Sons, 2012.
2. Seidman, Data Mining with Microsoft SQL Server, Prentice Hall of India,2016.

| CSE-418 | Parallel Computing | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **3** | **0** | **0** | **3** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **To enable students to compare various architectural taxonomies and design paradigms of parallel computers and computational models, parallelism approaches, performance metrics and techniques to parallelize and schedule loops and their programming constructs.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Classify various synchronous and asynchronous paradigms of parallel computing as well as identify some of the taxonomies for architectural classification of parallel computers. | | | | | | |
| **CO2** | Compare various parallel computation models and approaches and describe different performance metrics in parallel computers. | | | | | | |
| **CO3** | Distinguish shared memory and distributed memory multiprocessors and explainvarious parallel programming models and relative advantages and disadvantages of interconnection networks based on network parameters for reliable connections and achieving efficient speed. | | | | | | |
| **CO4** | Examine various techniques of parallelizing loops and sequential programs and scheduling. | | | | | | |

## Unit-I

**Introduction**: The state of computing, system attributes to performance, Paradigms of parallel computing: Synchronous – Vector/ Array, SIMD, systolic, Asynchronous- MIMD, reduction paradigm.

**Hardware Taxonomy**: Flynn’s classification, Feng’s classification, handler’s classification. **Software taxonomy**: Kung’s taxonomy.

## Unit-II

**Abstract parallel computational models**: combinational circuits, sorting network, PRAM models, VLSI complexity model, Interconnections RAMs, Parallelism approaches- data parallelism, control parallelism, Conditions of parallelism: Data, control and resource dependencies, Hardware and software parallelism.

**Performance metrics**: Laws governing performance measurements, Metrics- speedups, efficiency, utilization, communication overheads, single/ multiple program performances.

## Unit-III

**Parallel processors**: taxonomy and topology: shared memory multi processors, distributed memory multicomputer, static and dynamic interconnections.

**Parallel programming**: shared memory programming, distributed memory programming, object orientedprogramming, data parallel programming, functional and data flow programming.

## Unit-IV

**Scheduling and parallelization**: Loop parallelization and pipelining-Loop transformation theory,parallelization and wave fronting, tiling and localization, software pipelining, Scheduling parallel programs, program partitioning and scheduling: Grain size, latency, grain packing and scheduling, loop scheduling, Parallelization of sequential programs.

### Text Books

1. Kai Hwang and Naresh Jotwani, Advanced Computer Architecture, Second Edition,McGraw Hill, New Delhi, India, 2012.
2. M.J. Quinn, Parallel Computing: Theory and Practice, Second Edition, McGraw Hill,New Delhi, India, 2008.
3. D.Sima, T.Fountain, P.Kasuk, Advanced Computer Architecture-A Design spaceApproach, Pearson Education,India, 2009.

### Reference Books

1. J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative approach, 5th Edition, Morgan Kaufmann/Elsevier-India.
2. T.G.Lewis, Parallel Programming: A machine Independent approach, IEEE Computer Society Press,Los Alamitos, 1994.
3. T.G.Lewis and H. El-Rewini, Introduction to parallel computing, Prentice Hall, New Jersey, 1998.

| **CSE-420** | **Cloud Computing** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **03 Hrs.** |
| **Purpose** | **To familiar the concepts of cloud services and storageto deploy various resources and arbitrary software.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Facilitate the basic usage and applicability of computing paradigm. | | | | | | |
| **CO2** | Explore various cloud service and deployment models to utilize different cloud services. | | | | | | |
| **CO3** | To get enabled for various data, scalability &cloud services in order to get efficient database for cloud storage. | | | | | | |
| **CO4** | To deal with various security threats and their controlling mechanism for accessing safe cloud services. | | | | | | |

## Unit-1

Overview of Computing Paradigm: Recent trends in Computing, Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing, evolution of cloud computing, Business driver for adopting cloud computing.

Cloud Computing (NIST Model), History of Cloud Computing, Cloud service providers, Properties, Characteristics & Disadvantages, Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Role of Open Standards.

## Unit-2

Cloud Computing Architecture: Cloud computing stack, Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services, Service Models (XaaS) - Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud.

## Unit-3

Service Management in Cloud Computing: Service Level Agreements (SLAs), Billing & Accounting, comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling: Benefitting enormously, Managing Data- Looking at Data, Scalability & Cloud Services, Database & Data Stores in Cloud, Large Scale Data Processing.

Case study: Eucalyptus, Microsoft Azure, Amazon EC2.

## Unit-4

Cloud Security: Infrastructure Security, Network level security, Host level security, Application level security, Data security and Storage, Data privacy and security Issues, Jurisdictional issues raised by Data location, Identity & Access Management, Access Control, Trust, Reputation, Risk, Authentication in cloud computing, Client access in cloud, Cloud contracting Model, Commercial and business considerations.

### Text Books

1. Barrie Sosinsky, Cloud Computing Bible, Wiley-India, 2010.
2. Rajkumar Buyya,James Broberg, Andrzej M. Goscinski,Cloud Computing: Principles and Paradigms, Wiley, 2011.

### Reference Books

1. Nikos Antonopoulos, Lee Gillam, Cloud Computing: Principles, Systems and Applications, Springer, 2012.
2. Ronald L. Krutz, Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley-India, 2010.

| CSE-422 | Natural Language Processing | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | To provide the understanding of the mathematical and linguistic foundations underlying approaches to the various areas in NLP. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Be familiar with syntax and semantics in NLP. | | | | | | |
| **CO2** | To implement various concepts of knowledge representation using Prolog. | | | | | | |
| **CO3** | To classify different parsing techniques and understand semantic networks. | | | | | | |
| **CO4** | To identify/explain various applications of NLP. | | | | | | |

### Unit-1

**Fundamental components of Natural Language Processing**: Lexicography,syntax, semantics, prosody, phonology, pragmatic analysis, world knowledge.

**Knowledge Representation schemes**: Semantic net, Frames, Conceptual Dependency, Scripts.

### Unit-2

**Representing knowledge using rules:** Logic Programming, Introduction to LISP and Prolog, Rules based deduction systems, General concepts in knowledge acquisition.

**Syntax Analysis:** Formal Languages and grammars, Chomsky Hierarchy, Left- Associative Grammars, ambiguous grammars, resolution of ambiguities.

### Unit-3

**Computation Linguistics:** Recognition and parsing of natural language structures- ATN and RTN, General Techniques of parsing- CKY, Earley and Tomitas algorithm.

**Semantics:** Knowledge representation, semantics networks logic and inference pragmatics, graph models and optimization.

### Unit-4

**Applications of NLP:** Intelligent work processor, Machine translation, user interfaces, Man-Machine interfaces, natural language querying, tutoring and authoring systems, speech recognition, commercial use of NLP.

**Text Books:**

1. Daniel Jurafsky, James H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", 2nd edition, Pearson Edu., 2013.
2. James Allen, "Natural Language Understanding", Pearson Education, Second Edition, 2003.

**Reference Books:**

1. Ivan Bratko, "Prolog: Programming for Artificial Intelligence", 3rd Edition, Pearson Education, Fifth Impression 2009.
2. G. Gazder, "Natural Language processing in prolog", Addison Wesley, 1989.

| **CSE-404** | **Mobile Apps Development** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Major Test** | **Minor Test** | **Total** | **Time** |
| **4** | **0** | **0** | **4** | **75** | **25** | **100** | **3 Hrs.** |
| **Purpose** | **To introduce the concepts of developing the mobile applications.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Be exposed to technology and Mobile apps development aspects. | | | | | | |
| **CO2** | Be competent with the characterization and architecture of mobile applications. | | | | | | |
| **CO3** | Appreciation of nuances such as native hardware play, location awareness, graphics, and multimedia. | | | | | | |
| **CO4** | Perform testing, signing, packaging and distribution of mobile apps. | | | | | | |

## Unit 1: Introduction to Mobility

Mobility landscape, Mobile platforms, Mobile apps development, Overview of Android platform, Setting up the Mobile App Development environment along with an Emulator.

App User Interface Designing – Mobile UI resources (Layout, UI elements, Drawable, Menu).

## Unit II: Building blocks of Mobile Apps

Activity- States and Life Cycle, Interaction amongst Activities.

App functionality beyond user interface - Threads, Async task, Services – States and Life Cycle, Notifications, Broadcast receivers, Content provider.

## Unit III: Sprucing up Mobile Apps

Graphics and animation – Custom views, Canvas, Animation APIs, Multimedia – Audio/Video playback and record, Location awareness.

Native data handling –file I/O, Shared preferences, Mobile databases such as SQLite, and Enterprise data access (via Internet/Intranet).

## Unit IV: Testing Mobile Apps

Debugging mobile apps, White box testing, Black box testing, and test automation of Mobile apps, JUnit for Android.

**Text Books:**

1. Barry Burd, *Android Application Development All in one for Dummies*, Wiley publications, 2nd Edition 2015.
2. Android Developer Fundamentals Course– Concepts (Learn to develop Android applications) Concepts Reference *Developed by Google Developer Training Team, 2016.*
3. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design, and Development, Prentice Hall, 2004.
4. Rick Boyer, Kyle Mew,Android Application Development Cookbook - Second Edition,2016.

**Reference Books:**

1. Carmen Delessio, Lauren Darcey, Teach Yourself Android Application Development In 24 Hours , SAMS, 2013.
2. Brian Fling, Mobile Design and Development, O'Reilly Media, 2009.
3. Maximiliano Firtman, Programming the Mobile Web, O'Reilly Media, 2010.
4. Christian Crumlish and Erin Malone, Designing Social Interfaces, O'Reilly Media, 2009.
5. Jerome F. DiMarzio, Beginning Android Programming with Android Studio, 4th edition, 2016.
6. Max Lemann ,Android Studio: App Development on Android 6, 2016.

| CSE-406 | Mobile Apps Development Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **-** | **-** | **2** | **1** | **40** | **60** | **100** | **3 Hrs.** |
| **Purpose** | Design and Implement various mobile applications using emulators and learn how to Deploy applications to hand-held devices. | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | Know the components and structure of mobile application development frameworks for Android based mobiles. | | | | | | |
| **CO2** | Understand how to work with various mobile application development frameworks. | | | | | | |
| **CO3** | Learn the basic and important design concepts and issues of development of mobile applications. | | | | | | |
| **CO4** | Understand the capabilities of mobile devices. | | | | | | |

**List of Practicals:**

1. Develop an application that uses GUI components, Font and Colours
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Implement an application that implements Multi threading
6. Develop a native application that uses GPS location information.
7. Implement an application that writes data to the SD card.
8. Implement an application that creates an alert upon receiving a message.
9. Write a mobile application that creates alarm clock.
10. Develop a sign-in page with appropriate validation.
11. Develop a real life application that makes use of database.

**Note: At least 5 to 10 more exercises are to be given by the teacher concerned.**

| CSE-408 | Computer Hardware & Troubleshooting Lab | | | | | | |
|---|---|---|---|---|---|---|---|
| **Lecture** | **Tutorial** | **Practical** | **Credit** | **Minor Test** | **Practical** | **Total** | **Time** |
| **0** | **0** | **2** | **1** | **40** | **60** | **100** | **3 Hrs.** |
| **Purpose** | **To study the current personal computer hardware including personal computer assembly upgrading, setup configuration and troubleshooting.** | | | | | | |
| **Course Outcomes (CO)** | | | | | | | |
| **CO1** | To understand the fundamental hardware components that makes up a computer's hardware and the role of each of these components. | | | | | | |
| **CO2** | Assemble/setup and upgrade personal computer hardware. | | | | | | |
| **CO3** | Perform installation, configuration, and upgrading of microcomputer hardware and software. | | | | | | |
| **CO4** | Diagnose and troubleshoot microcomputer systems hardware and software, and other peripheral equipment. | | | | | | |

**List of Practicals:**

1. To make the comparative study of various motherboards.
2. To study various cables used in computer communication.
3. To study various connections and ports used in computer communication.
4. To study various cards used in a computer System like Ethernet, sound, video card etc.
5. To study different microprocessor like P-IV, dual core, i3, i5, i7 etc.
6. To study SMPS and UPS.
7. To study rotational and loading mechanisms of the following drives:(Floppy disk drive, Hard disk, CD ROM,CD-R/RW,DVD-ROM, DVD recordable drives, DUAL LAYER DVD-R/W)
8. To study monitor and its circuitry (CRT (Cathode Ray Tube), LCD (Liquid Crystal Display), LED (Light-Emitting Diodes), Plasma (OLED).
9. To study different types of printers and its installation.
10. To study working of keyboard and mouse.
11. To assemble a PC and trouble shooting.
12. To install different Operating System and install different hardware components.

**Text Books:**

1. How Computers WorkBy, Ron White and Timothy Edward Downs, 10th Revised edition, Pearson Education, 2014.
2. Upgrading and Repairing PCs, Scott Mueller,22nd Edition,Que Publishing, 2015.
3. Learning PC Hardware, Ramesh Bangia, Khanna Book Publishing, 2nd revised edition, 2012.

**Reference Books:**

1. Pc Hardware: The Complete Reference 1st Edition,Craig Zacker, McGraw Hill Education, 1st edition, 2001.
2. Modern Computer Hardware Course, ManaharLotia, Pradeep Nair, PayalLotia, BPB Publications, 2nd Revised Edition, 2007.