B.Tech: 3\textsuperscript{rd} Semester

<table>
<thead>
<tr>
<th>Code</th>
<th>Course</th>
<th>Theory</th>
<th>Practical</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 2404</td>
<td>Mathematical Methods</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>CSE 2403</td>
<td>Computer Organization</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>IT 2401</td>
<td>Data Structures</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>ECE 2406</td>
<td>Digital Logic Design</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>CSE 2404</td>
<td>Discrete Structures and Graph Theory</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>IT 2203</td>
<td>Data Structures Lab</td>
<td>-</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CSE 2205</td>
<td>Digital Logic Design Lab</td>
<td>-</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>20</strong></td>
<td><strong>6</strong></td>
<td><strong>24</strong></td>
</tr>
</tbody>
</table>

B.Tech 4\textsuperscript{th} Semester

<table>
<thead>
<tr>
<th>Code</th>
<th>Course</th>
<th>Theory</th>
<th>Practical</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSE 2406</td>
<td>Data Communication Systems</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>IT 2405</td>
<td>Database Management Systems</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>IT 2402</td>
<td>Object Oriented Programming through Java</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>CSE 2407</td>
<td>Operating Systems</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>CSE 2408</td>
<td>Theory of Computation</td>
<td>3+1*</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>IT 2207</td>
<td>Database Management Systems Lab</td>
<td>-</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>IT 2204</td>
<td>Object Oriented Programming through Java Lab</td>
<td>-</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>20</strong></td>
<td><strong>6</strong></td>
<td><strong>24</strong></td>
</tr>
</tbody>
</table>

*Tutorial
Course Title: MATHEMATICAL METHODS  
Course Code: L T P C
3 1 0 4

Course Objectives: The course content enables students to:

- Solve linear system of equations,
- Apply numerical techniques in Engineering problems.
- Apply the concept of curve fitting for the data given.
- Learn the knowledge of finding the area bounded by curves using numerical methods.
- Know the methods of finding the roots of transcendental equations.

Course Outcomes: At the end of the course students are able to:

- Apply Linear system of equations in animation and simulation techniques used in applications like Gaming and Prototype Modeling for incorporating simulated physical phenomena such as quality, collision, friction, fluid flow etc.
- Understand the use of the concept of linear equations in the study of “Computer Vision”, used in areas such as Artificial Intelligence, Industrial Robotics, Human Computer Interaction, Video Tracking etc.
- Understand the use of the concepts of Linear system of equations, Eigen values & vectors in the study of Pattern Recognition, Document Clustering and Classification, etc.
- Apply the concepts of curve fitting in data clustering and classification studied in Data Mining
- Solve problems using Lagrange method of interpolation
- Use Newton divided difference interpolation in Information Security for securing the message (message is converted into polynomial)
- Apply Lagrange interpolation techniques in group key generation, key agreement and key management
- Use interpolation with evenly spaced points in 3D Computer Graphics
- Apply numerical integration techniques in applications such as Simulation for Prototyping
- Simulate real world phenomena and events
- Analyze and solve science & engineering problems applying the mathematical methods.

UNIT – I  
15 Hrs 
Eigen values - Eigen vectors – Properties – Cayley-Hamilton Theorem - Inverse and powers of a matrix by using Cayley-Hamilton theorem.
Quadratic forms- Reduction of quadratic form to canonical form – Rank - Positive, negative definite - semi definite - index – signature.
UNIT – II  
15 Hrs
Curve fitting: Fitting a straight line –Second degree curve-exponential curve-power curve by method of least squares.

UNIT-III  
15 Hrs

UNIT – IV  
15 Hrs

Text Book :

Reference Books :
Course Title: COMPUTER ORGANIZATION

Course Objectives: The course content enables students to:

- Understand how a computer system performs tasks by executing different micro-operations.
- Understand the basic organization of Computer system and its operation.
- Understand the instruction formats with different addressing modes, used by the CPU for instruction processing.
- Know how control unit generates signals for carrying out instruction execution.
- Understand and analyze how the CPU performs basic arithmetic operations.
- Understand the organization of the memory system and its effect on performance of the computer.
- Understand how data transfer takes place among the various peripherals in the computer system.
- Know the different forms of concurrent processing and its effect on execution speed of the computer.

Course Outcomes: At the end of the course students are able to:

- Know the different components of the computer system, their functions and their interconnections.
- Know the various instruction formats that the processor follows.
- How the control unit generates control signals to execute a particular instruction.
- Understand how the memory system can be suitably designed to improve the performance of the computer.
- Know how high speed computers can be designed by using the pipelining and multiprocessor concepts.

UNIT – I
COMPUTER FUNCTION AND REGISTER TRANSFER LANGUAGES: Computer types, Functional units, Register transfer language, Register transfer, Bus and memory transfers
MICRO-OPERATIONS: Arithmetic micro-operations, Logic micro-operations, Shift micro-operations, Arithmetic logic shift unit
BASIC COMPUTER ORGANIZATION AND DESIGN: Instruction codes, Computer registers, Computer instructions, Timing and Control, Instruction cycle, Memory-Reference instructions, Register-Reference instruction, Input-Output instruction, Interrupts, Design of basic computer

UNIT – II
CENTRAL PROCESSING UNIT: Stack organization, Instruction formats, Addressing modes, Data Transfer and manipulation, Program control, reduced instruction set computer, Complex instruction set computer
CONTROL UNIT DESIGN: Hardwired control unit design, Micro-programmed Control unit design, Control memory, Address sequencing, Micro-program example

UNIT-III 16Hrs
COMPUTER ARITHMETIC: Fixed point representation, Floating point representation, Addition and subtraction, Multiplication algorithms, Division algorithms, Floating point arithmetic operations
THE MEMORY SYSTEM: Memory hierarchy, Semiconductor RAM memories, Read-only memories, Cache memories, Performance considerations, Secondary storage, Virtual memories

UNIT-IV 14Hrs
INPUT-OUTPUT ORGANIZATION: Peripheral devices, Input-Output interface, Asynchronous data transfer, Modes of transfer, Priority interrupt, Direct memory access, Input-Output processor (IOP)
PIPELINE AND VECTOR PROCESSING: Parallel processing, Pipelining, Arithmetic pipeline, Instruction pipeline, Vector processing, Multiprocessors-loosely coupled and tightly coupled

Text Books:

Reference Books:
Course Title: DATA STRUCTURES

Course Objectives: The course content enables students to:
- Understand various data structures and their importance in algorithm design.
- Implement the linear data structures stack and queue using array.
- Implement various forms of linked-lists.
- Implement various Hashing techniques using appropriate data structures.
- Demonstrate different Binary Tree Traversals.
- Demonstrate Binary Search Tree operations.
- Understand the advantages of height balanced trees.
- Implement graphs using either adjacency matrix or adjacency list.

Course Outcomes: At the end of the course students are able to:
- Create and manage data structures for developing real world applications.
- Use algorithms and data structures in well-written modular code in an adequate and optimal way.
- Apply the data structure array for various list processing operations such as searching and sorting.
- Decide to use array or linked-list for solving a given problem.
- Differentiate between linear and non-linear data structures.
- Use appropriate Data Structure for implementation of various algorithms in subsequent course of the study.

UNIT – I
Data Structures and their importance in algorithm design: linear and non-linear data structures
Searching: Linear search and Binary Search
Sorting: Selection sort, bubble sort, insertion sort
Stacks: Basic operations, Implementation using array, Applications – Recursion, Infix to Postfix conversion
Queues: Basic operations, Linear queue, Circular Queue, implementation using array

UNIT – II
Linked Lists: Basic operations, singly linked lists, doubly and circular linked lists
Hashing: Hashing Functions, Open hashing(chaining), closed hashing(open addressing – linear probing, quadratic probing, double hashing), rehashing

UNIT-III
Trees: Binary Trees, Terminology, Representation of Binary Trees using arrays and linked lists, Binary tree traversals, Creation of binary tree from in-order, pre-order and post-order traversals
Binary Search Trees: Searching, Insertion and deletion in BST
Heaps: Max-heap/ Min-heap, insertions and deletions, Heap sort
UNIT-IV

**Balanced Trees:** AVL trees, Height of AVL Tree, Balancing AVL tree by rotations, insertions and deletions

**B-Trees:** B-Tree of order $M$, insertions and deletions

**Graphs:** Graph Traversals (BFS & DFS)

**Text Books:**
2. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Pearson Education

**Reference Books:**
2. Data Structures with C, Seymour Lipschutz, McGraw Hill
Course Title: DIGITAL LOGIC DESIGN  
Course Code: ECE 2406

**Course objectives:** Students undergoing this course are expected to:

- Understand the different number system, its conversions and binary arithmetic.
- KNOW the fundamentals of Boolean algebra and theorems, Karnaugh maps including the minimization of logic functions to SOP or POS form.
- Analysis of logic circuits and optimization techniques to minimize gate count, signals, IC count, or time delay.
- Strengthen the principles of logic design and use of simple memory devices, flip-flops, and sequential circuits.
- Fortify the documentation standards for logic designs, standard sequential devices, including counters and registers.
- understand the logic design of programmable devices, including PLDs
- understand RAMS, and ROMS including its sequencing and control

**Course outcomes:** After undergoing the course students will be able to:

- Differentiate between analog and digital representations.
- Convert a number from one number system to its equivalent in of the other Number system.
- Understand the difference between BCD and straight binary.
- Implement logic circuits using basic AND, OR and NOT gates.
- Use De-Morgan’s theorem to simplify logic expressions.
- Describe the concept of active LOW and active HIGH logic signals.
- Use Boolean algebra and K-map as tool to simplify and design logic circuits.
- Design simple logic circuits without the help of truth tables.
- Construct and analyze the operation of flip-flop and troubleshoot various types of flip-flop circuits.
- Understand the internal logic diagrams of Integrated circuits
  Design various types of sequential circuits like counters, state machines etc
UNIT- I
NUMBER SYSTEMS AND BOOLEAN ALGEBRA:

Review of number systems, conversion of numbers from one radix to another radix, complement representation of negative numbers-binary arithmetic, 4-bit codes: BCD, Excess-3, Floating point representation(IEEE 754 Standard), Fixed point representation, Basic logic operations. Basic theorems and properties of Boolean Algebra, switching functions, Canonical and Standard forms-Algebraic simplification digital logic gates, universal gates and Multilevel NAND/NOR realizations, Generation of self dual functions. Gray code, error detection and error correction codes, parity checking even parity, odd parity, Hamming code

UNIT- II
BOOLEAN FUNCTION MINIMIZATION AND COMBINATIONAL LOGIC CIRCUITS:

Minimization of switching functions using K-Map up to 6-variables, Tabular minimization, minimal SOP and POS Realization, Problem solving using K-map such as code converters binary Multiplier. Half adder, Full adder, full subtractor, Ripple carry adder, Carry look ahead adder, Multiplexer, De-Multiplexer Encoder, Priority encoder, Decoder, MUX Realization of switching functions Parity bit generator.

UNIT- III
PLDs AND SEQUENTIAL CIRCUITS – I:
Basic PLD’s-ROM, PROM, PLA, PAL, Realization of Switching functions using PLD’s, comparison of PROM,PLA,and PAL.Classification of sequential circuits (synchronous and asynchronous): basic flip-flops, truth tables and excitation tables (NAND RS latch, NOR RS latch, RS flip-flop. JK flip-flop, T flip-flop, D flip-flop with reset and clear terminals).Conversion of flip-flops.

UNIT- IV
SEQUENTIAL CIRCUITS – II AND SM CHARTS:

Text Books:

Reference Books:
2. Modern Digital Electronics by RP Jain, TMH.
Department of Computer Science & Engineering
B.Tech - 3rd Semester

SYLLABUS
(Applicable for 2012 and 2013 admitted batches)

Course Title: DISCRETE STRUCTURES AND GRAPH THEORY  Course Code:  L     T     P     C

3     1     0      4

Course Objectives: The course content enables students to:

• Comprehend the structure of statements (and arguments) involving predicates and quantifiers. Students can able to apply logic on statements.
• Understand the applications of graph theory to various practical problems.
• Understand the basic properties of sets and relations.
• Understand proofs of various properties in group theory.
• Know the concept of permutations and combinations.
• Know how to solve a recursive problem.

Course Outcomes: At the end of the course students will be able to:

• Apply the concept of logical equivalence and its relationship to logic circuits and Boolean functions.
• Use concepts of graph theory to provide solutions for routing applications in computer networks.
• Use concepts of functions for data retrieval in database applications.
• Apply concepts of counting and probability in data mining.
• Apply the recurrence relation for analyzing recursive algorithms.

UNIT – I  16 Hrs

UNIT – II  14 Hrs
Graph Theory: Representation of Graph, DFS, BFS, Planar Graphs, Chromatic Number, Graph Theory and Applications, Basic Concepts of Isomorphism and Sub graphs, Euler circuits, Hamiltonian graphs, Spanning Trees, Minimal spanning tree,

UNIT-III  14 Hrs
Algebraic structures: Algebraic systems Examples and general properties, Semi groups and Monoids, Groups, Sub group

UNIT-IV  16 Hrs
Elementary Combinatorics: Basis of counting, Permutations & Combinations with repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion–Exclusion, Pigeon hole principle and its application.

Recurrence Relation: Generating Functions, Function of Sequences, Calculating Coefficient of generating function, Recurrence relations, Solving recurrence relation by substitution and Generating functions.

Text Books:
1. Discrete Mathematical Structures with applications to computer science Tremblay J.P. & P.Manohar, TMH

Reference Books:
1. Discrete Mathematics with Applications, Thomas Koshy, Elsevier
2. Discrete Mathematical structures Theory and application-Malik &Sen
3. Graph theory with applications to engineering and computer science, by Narsingh Deo – PHI
Course Title: DATA STRUCTURES LAB

Implement the following experiments:

1. Write programs to perform Linear search and Binary search for a Key value in a given list.
2. Implement any two sorting techniques using an appropriate data structure
3. Implement Stacks and Queues using arrays
4. Implement Singly Linked Lists.
5. Implement Doubly Linked Lists
6. Implement Stacks and Queues using linked lists
7. Implement Open Hashing (Chaining) and Closed Hashing (Linear Probing) using appropriate data structures
8. Write a program to demonstrate different Binary Tree Traversals.
9. Write a program to demonstrate BST operations
10. Write a program to demonstrate AVL tree operations
11. Write a program to demonstrate Heap operations
12. Write a program to demonstrate different Graph Traversals
Course Title: DIGITAL LOGIC DESIGN LAB

Course Code:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Implement the following experiments:

1. Number base conversions
2. 4-bit word Parity generator/checker
3. Realization of Logic gates and Verification of Truth Tables
4. Verification of Boolean Laws and Theorems using Truth tables
5. Realization of basic gates using Universal gates
6. Construction of half-adder, half-subtractor, full-adder, full-subtractor and verification of their truth tables
7. Verification of the properties of decoders and encoders
8. Verification of the functioning of Multiplexer and De-multiplexers
9. Realization of 4-Bit comparator using IC.
10. Realization of Flip-Flops using IC’s.
11. Verification of functioning of Basic Shift Register (SR), SI/SO SR, SI/PO SR, PI/SO SR, PI/PO
12. Realization of Up/Down, Modulo - 5, Modulo – 10 counters
Department of Computer Science & Engineering

B.Tech- 4th Semester

SYLLABUS

(Applicable for 2012 and 2013 admitted batches)

Course Title: DATA COMMUNICATIONS SYSTEMS Course Code:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Course objectives: The course content enables students to:

- Understand fundamentals of Standards and n/w architecture and Types of Data Transmission and Modulation systems
- Understand the building blocks of an Optical Fiber system
- To explore the terminology used in optical fibers.
- Analyzing PCM and their types and T CARRIERS and TDM, FDM
- Understand the Electromagnetic Waves and Satellite Communications Systems
- Designing the Telephone Circuit with various arrangements
- Acquire knowledge about Cellular Telephone Systems and Digital Cellular Telephone
- Emphasize data on various error detection and correction techniques.
- Explore the concepts of Character –and Bit- Oriented Protocols,
- Explore Asynchronous and Synchronous Data – Link Protocols and HDLC

Course Outcomes: At the end of the course students will be able to:

- Acquire knowledge of fundamental concepts of data transmissions standards and to learn the fundamental digital techniques for Communication.
- Understand the design, operation and capabilities of optical fiber systems.
- Acquire knowledge of various digital transmissions (PCM) & multiplexing of FDM and TDM
- Design a system, component or process as per needs and specification.
- Develop applications by analyzing the requirements of software.
- Understand the technical specifications of electronic communication design.
- Know the concepts of Mobile Communications.
- Explore the knowledge of various error detection and correction techniques.
- Design and analyze the experiment and interpret data.
- Acquire knowledge of Data link protocols like Character –and Bit- Oriented Protocols and HDLC.

UNIT – I 16Hrs

INTRODUCTION TO DATA COMMUNICATIONS AND NETWORKING:
Standards Organizations for Data Communications, Layered Network Architecture, Open Systems Interconnection, Serial and parallel Data Transmission.
Signals, Noise, Modulation and De-Modulation: Signal Analysis, Electrical Noise, M-ary Encoding, Analog and Digital Modulation systems.

UNIT – II                        15hrs
DIGITAL TRANSMISSION:
Pulse Modulation, Pulse code Modulation, Linear Versus Nonlinear PCM Codes, Delta Modulation, PCM and Differential PCM.
MULTIPLEXING and T CARRIERS: Time-Division Multiplexing, T1 Digital Carrier System, North American Digital Multiplexing Hierarchy, TCarrriersystems, Frequency-Division Multiplexing.

UNIT-III                     15Hrs
TELEPHONE INSTRUMENTS AND SIGNALS:
The Subscriber Loop, Standard Telephone Set, Basic Telephone Call Procedures, Cordless Telephones, Paging systems.

UNIT-IV                     14Hrs
DATA COMMUNICATIONS CODES, ERROR CONTROL, AND DATA FORMATS:
Data Communications, Character Codes, Error Control, Error Detection and Correction.
Data Communications Equipment: Digital Service Unit and Channel Service Unit, Voice- Band Modern Block Diagram, Voice- Band Data Communication Modems.

Text Books:
1. Introduction to Data Communications and Networking, Wayne Tomasi, Pearson Education.

Reference Books:
GMR Institute of Technology
An Autonomous Institute Affiliated to JNTUK, Kakinada

Department of Computer Science & Engineering
B.Tech- 4th Semester

SYLLABUS
(Applicable for 2012 and 2013 admitted batches)

Course Title: DATABASE MANAGEMENT SYSTEMS

Course Objectives: The course content enables students to:

- Understand the differences between File system and DBMS, Data Models and database system structure.
- Know how to use the integrity constraints over the relations and expressive power of Algebra and calculus.
- Learn the query language features which are the core of SQL’s DML, Join operations and Triggers.
- Learn normalization procedure to eliminate the redundancy in the databases.
- Know the concept of the transaction management which is the foundation for concurrent execution and recovery from the system failure in a DBMS.
- Learn the recovery techniques for managing the database effectively and avoid the data lose.
- Know how to arrange the records in a file when the file is stored on the external storage.

Course Outcomes: At the end of the course students will be able to:

- Identify and define the data models needed to design a database.
- Create conceptual and logical database design for Large enterprises.
- Apply Integrity constraints over the relations.
- Apply normalization process on existing database for eliminating redundancy.
- Apply the recovery techniques for managing the database effectively to avoid the data lose.

UNIT I 15Hrs

Introduction to DBMS: Database System Applications, database System Vs file System, View of Data, Data Abstraction, Instances and Schemas, data models, the ER Model, Relational Model, Network model, Hierarchy model. Database Languages: DDL, DML, DCL, DBMS architecture.

Database Design: Introduction to database design, ER Model, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual design for large enterprises.

UNIT II 15Hrs

Introduction to the Relational Model: Integrity constraints, Relational Algebra, Selection and projection set operations, renaming, Joins, Division, Relational calculus: Tuple relational Calculus, Views.
SQL Queries: Form of Basic SQL Query, Introduction to Nested Queries, Correlated Nested Queries, Set Comparison Operators, Aggregative Operators – NULL values, Outer Join, Logical connectivity’s, AND, OR and NOT, Triggers.

UNIT III 15Hrs


Transactions: Transaction State, ACID properties of transaction, serial schedule, parallel schedule, conflicts in concurrent Executions, Serializability, Recoverability, performance of locking, transaction support in SQL.

UNIT IV 15Hrs

Concurrency Control: Introduction to Lock Management, Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques, Concurrency without Locking.

Crash Recovery: Introduction to ARIES, the Log, other recovery related structures, the Write-Ahead Log Protocol, Checkpointing – recovering from a system.

Data on External Storage: File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure,

Database Security: Threats and risks, Database access control, Types of privileges,

TEXT BOOKS:


REFERENCES:

3. Introduction to Database Systems, C.J.Date Pearson Education
Course Title: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Course Objectives: The course content enables students to:

- Understand fundamentals of object-oriented concepts through Java.
- Understand how Java achieves platform independence using concept of bytecode.
- Extend existing code and develop new application to demonstrate code reusability.
- Keep the related class of code together to create a package and import the same for future application development.
- Implement multiple inheritances using interface concept.
- Explore concepts of concurrent programming by using multi threading.
- Handle runtime errors through exception handling mechanism.
- Provide graphical user interface for their application programs.
- Write applications that handle user interactions through various peripheral devices.

Course Outcomes: At the end of the course students will be able to:

- Know the concepts of classes, objects, members of a class and the relationships among them.
- Implement Applications using Packages.
- Handle runtime errors using Exceptions handling mechanism.
- Develop application for concurrent processing using Thread concepts.
- Design interactive applications for use on internet.
- Design applets that take user response through various peripheral devices such as mouse and keyboard by event handling mechanism.

UNIT-I
Introduction to Java: 14Hrs
Overview of Object Oriented Programming principles, Importance of Java to the Internet, Bytecode, Methods, classes and instances. Data types, arrays, control statements, simple java program. Classes and Objects – constructors, methods, access control, this keyword, overloading methods and constructors, garbage collection.

UNIT-II
Packages: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, Member access rules.
Interface: Defining an interface, differences between classes and interfaces, implementing interface, variables in interface and extending interfaces.
UNIT- III  
**Exception handling:** Concepts and benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built-in and User Defined Exceptions,  
**Multithreading:** Definition thread, thread life cycle, creating threads, synchronizing threads, daemon threads.

UNIT IV  
**Applets:** Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets, The AWT class hierarchy, user interface components- labels, button, Text components.  
**Event Handling:** Events, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. Compare basic AWT components with swing components. More user interface components - canvas, scrollbars, check box, choices, lists panels – scrollpane, dialogs, menubar, layout manager types.

**Text Books:**

**Reference Books:**
2. Programming with Java A Primer, E.Balaguruswamy Tata McGraw Hill Companies  
Department of Computer Science & Engineering  
B.Tech- 4th Semester  
SYLLABUS  
(Applicable for 2012 and 2013 admitted batches)

Course Title: OPERATING SYSTEMS  
Course Code:  
L    T      P    C  
3     1      0    4

Course Objectives: The course content enables students to:

• Understand Objectives, Functions, Services of Operating Systems and Learn the Concepts of Process and Process Scheduling
• Understand issues related to Process Synchronization and focus on principles of Deadlock and related problems of Starvation
• Comprehend the mechanisms used in Memory Management and Virtual Memory.
• Understand the concepts of File System and Disk Scheduling

Course Outcomes: At the end of the course students will be able to:

• Understand the various concepts of process
• Implement concurrency mechanisms
• Implement Bankers Algorithms to handle deadlocks
• Design and analyze mechanisms used in memory management
• Develop processor scheduling, Paging technique Algorithms

UNIT – I

UNIT – II

UNIT – III
Process Synchronization: Introduction to Process synchronization, The Critical-Section Problem, Peterson’s Solution, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization using Semaphores

Memory Management: Introduction to Memory Management, Swapping, Contiguous Memory Allocation, Paging, Segmentation.
Virtual Memory Management: Introduction to Virtual Memory Concept, Demand Paging, Copy on Write, Page Replacement Algorithms, Allocation of Frames, Thrashing.
UNIT-IV


Text Books:

Reference Books:
2. Operating System A Design Approach-Crowley, TMH.
Department of Computer Science & Engineering
B.Tech- 4th Semester

SYLLABUS
(Applicable for 2012 and 2013 admitted batches)

Course Title: THEORY OF COMPUTATION  
Course Code: L T P C
3 1 0 4

Course Objectives: The course content enables students to:
• Understand the concept of Set Operations, Cardinality and Denumerable sets. Acquire knowledge to design, specify and test deterministic and nondeterministic finite automata that recognize regular languages.
• Understand tuple descriptions of Mealy and Moore finite-state machines, State diagrams, state tables and their relationship.
• Acquire the idea to write regular expressions and regular grammars that produce regular languages.
• Identify non-regular languages using the Pumping Lemma.
• Acquire Knowledge of Formal Languages, Grammars, Derivations and parse trees.
• Understand the concept of transforming context-free grammars by removing useless productions, lambda productions and unit productions.
• Acquire the idea to design, specify and test grammars and also deterministic/nondeterministic pushdown automata that recognize context-free languages.
• Understand the knowledge to design, specify and test Turing Machines that recognize recursively enumerable languages.

Course Outcomes: At the end of the course students will be able to:
• Know the regular expressions and regular grammars that produce regular languages.
• Analyze and solve Chomsky and Greibach normal forms.
• Design deterministic and nondeterministic finite automata that recognize regular languages.
• Develop push down automata for context free grammars.
• Design Turing machines those recognize recursively enumerable languages.

UNIT – I 16Hrs
Finite automata: Finite state machine, definitions, finite automation model, acceptance of strings and languages, Deterministic finite automaton and non-deterministic finite automaton, transition diagrams. NFA with  \( \varepsilon \) transitions, Equivalence between NFA with and without  \( \varepsilon \) transitions, NFA to DFA conversion, minimization of FSM, equivalence between two FSM’s Finite Automata with output-Moore and Mealy machines.

UNIT – II 14Hrs
Regular languages: Regular sets, regular expressions, identity rules, Constructing finite Automata for a given regular expressions, Conversions of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets (proofs not required).Regular grammars-right linear and left linear grammars, interconversion between regular grammar and FA, Context free grammar, derivation trees, right most and leftmost derivation of strings.
UNIT – III  
**Context Free Grammars and PDA:** Ambiguity in context free grammars. Minimization of Context Free Grammars, Chomsky normal form, Greibach normal form, Enumeration of properties of CFL (proofs omitted). Push down automata, definition, model, acceptance of CFL, acceptance by final state and acceptance by empty state. Inter conversion CFL and PDA.

UNIT – IV  
**Turing Machine:** Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages, types of Turing machines, Chomsky hierarchy of languages, linear bounded automata and context sensitive language, undecidability of posts Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

**Text book:**
1. Introduction to Automata theory Language and Computation by Jeffery D. Ullman and John E. Hopcroft, Narosa Publishing House.
2. Introduction to computer theory, Daniel I.A cohen, John Wiley.

**REFERENCES:**
1. Theory of computer science –Automata language and computation – Mishra and Chandrashekar, 2nd Edition, PHI.
Course Title: DATABASE MANAGEMENT SYSTEMS LAB

Course Code:

Implement the following experiments:

1. Execute Single line and Group functions for a table.
2. Implement Queries using GROUP BY & HAVING clauses, set operators and set comparison operators.
3. a) Execute DCL commands
   b) Creation and dropping of Views.
4. Create and manipulate various DB objects for a table.
5. a) Creation of simple PL/SQL program which includes declaration section, executable section
    and exception – Handling section
   b) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
6. Develop a program that includes the features of NESTED IF and CASE. The program can be extended
   using the NULLIF and COALESCE functions.
7. Develop a Program using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling,
   BUILT-IN Exceptions, User-defined Exceptions, RAISE- APPLICATION ERROR.
8. Develop a Program using creation of procedures, passing parameters IN and OUT of PROCEDURES.
9. Develop a Program using creation of stored functions, invoke functions in SQL Statements and write
   complex functions.
10. Develop a Program using packages.
11. Write PL/SQL procedure for an application using cursors.
12. Create sample triggers and assertions.
Course Title: OBJECTORIENTED PROGRAMMING THROUGH JAVA LAB

Course code: L T P C
0 0 3 2

Implement the following experiments:

Use JDK 1.5 or above on any platform e.g. Windows or Unix.

1. Write a Java program to demonstrate String handling methods.
2. Write a Java program for sorting a given list using inheritance concept.
3. Write a Java program for creating one base class for student personal details and inherit those details into the sub class of student Educational details to display complete student information.
4. Write a Java program to implement matrix operations using multidimensional arrays
5. Write a Java program that illustrates runtime polymorphism
6. Write a Java program, to demonstrate tokenizing given string/text using StringTokenizer class
7. Write a Java program to create a package which has classes and methods to read Student Admission details.
8. Write a Java program to define and handle Exceptions in the implementation of Program3.(also make use of throw, throws).
9. Write a Java program to create multiple threads for different calculator operations.
10. Write an Applet to draw various geometrical shapes
11. Write a Java program for handling mouse events.
12. Write a Java Program to design a Job Application/ Student Admission Form.
13. Write a Java program that works as a simple Calculator.