TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

B.TECH – COMPUTER SCIENCE & ENGINEERING

Course Structure R-20

SEMESTER III

S.No.	Class	Course Code	Name of the Subject	L	T	P	C
1	HS	CHSM1	Business Economics and Financial Analysis	3	0	0	3
2	BS	CBSM4	Mathematical Foundations of Computer Science	3	0	0	3
3	ES	CESLC1	Logic Circuits Design	3	0	0	3
4	PC	C53PC1	Database Management Systems	3	0	0	3
5	PC	C53PC2	Data Structures	3	0	0	3
6	PC	C53PC3	Operating Systems	3	0	0	3
7	ES	CESLC2	Logic Circuits Design Lab	0	0	2	1
8	PC	C53PC4	Database Management Systems Lab	0	0	2	1
9	PC	C53PC5	Data Structures Lab	0	0	2	1
10	MC	MC003	Cultural Activity	0	0	0	Satisfactory
Total Credits				**18**	**0**	**6**	**21**

Mandatory Course: Cultural Activity
The student should participate in culture activity (Music/Dance/Singing/etc.) conducted by the College,student should produce the participation certificate for clearing this course.

SEMESTER IV

S.No.	Class	Course Code	Name of the Subject	L	T	P	C
1	BS	CBSM3	Probability & Statistics	3	0	0	3
2	PC	C54PC1	Computer Organization & Architecture	3	0	0	3
3	PC	C54PC2	Software Engineering	3	0	0	3
4	PC	C54PC3	Design and Analysis of Algorithms	3	0	0	3
5	PC	C54PC4	Formal Languages & Automata Theory	3	0	0	3
6	PC	C54PC5	Object Oriented Programming through Java	3	0	0	3
7	PC	C54PC6	Object Oriented Programming through JavaLab	0	0	2	1
8	PC	C54PC7	Computer Organization & Architecture Lab	0	0	2	1
9	PC	C54PC8	Design and Analysis of Algorithms Lab	0	0	2	1
10	MC	MC004	Video with Social Messages	0	0	0	Satisfactory
Total Credits				**18**	**0**	**6**	**21**

Mandatory Course: Video with Social Messages
Student should make video with social messages. This has to be uploaded in the youtube.com, by maintaining the terms and conditions of youtube.com. Student should produce youtube.com link with screen shot for clearing this mandatory course.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
3/0/0 3

BUSINESS ECONOMICS AND FINANCIAL ANALYSIS (CHSM1)

Course Objectives:

1. To learn the basic business type of the organization.
2. To aquire the knowledge and impact of the economy on business firms.
3. To analyse the business from the financial perspective.
4. To know the financial position of the company.

Course Outcomes:

1. Analyze the total structure of the business & able to identify and classify the different types of business entities. L4
2. Analyze the demand & supply analyses with the help of various measures and types of Elasticity of demand. L5
3. Analyze the knowledge about production and cost analysis for product and services. L4
4. Demonstrate the fundamental concepts related to financial accounting. L3
5. Analyze the financial position by analyzing the financial statement of the Company through various ratios. L5

UNIT I

Introduction to Business and Economics

Business: Define Business, characteristics of business, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company.

Economics: Significance of Economics, Micro and Macro Economic Concepts and Importance of National Income, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist.

UNIT II

Demand Analysis

Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Steps in Demand Forecasting, Methods of Demand Forecasting.

UNIT III

Production, Cost, Market Structures & Pricing

Production Analysis: Production function, Law of returns to scale, Internal and External Economies ofScale.

Cost Analysis: Cost concepts, Types of costs, Break-Even Analysis (BEA)

Pricing: Types of pricing, product life cycle, **GST (Goods & Service Tax)**

Market Structures: Types of competition, Features of Perfect competition, Monopoly and Monopolistic competition, oligopoly.

UNIT IV

Financial Accounting

Accounting concepts and Conventions, Accounting Equation, Double-Entry system of

Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, and Preparation of Final Accounts.

UNIT V

Financial Analysis through Ratios

Concept of Ratio Analysis, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios (simple problems).

Text Books:

1. D. D. Chaturvedi, S. L. Gupta, Business Economics - Theory and Applications International Book House Pvt. Ltd. 2013.
2. Dhanesh K Khatri, Financial Accounting, Tata McGraw Hill, 2011.
3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, Managerial Economics, 2e, Tata McGraw Hill Education Pvt. Ltd. 2012.
4. Rakesh Garg, Sandeep Garg Hand book of GST in India.
5. A.R. Aryasri (2011) Managerial Economics and Financial Analysis, TMH, India.

Reference Books:

1. Paresh Shah, Financial Accounting for Management 2e, Oxford Press, 2015.
2. S.N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, Financial Accounting, 5e, Vikas Publications, 2013.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
3/0/0 3

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE (CBSM4)

Course Objectives:
To provide students with a thorough understanding of the mathematical principles and techniques that form the foundation of computer science.

Course Outcomes:
After learning the contents of this course, the student must be able to:

1. Apply mathematical logic to solve problems. L3
2. Analyze the assertions using predicate logic. L4
3. Analyze different properties of GCD using Division and Euclidean Algorithm. L4
4. Analyze the basic terminology of functions, relations, sets and demonstrate knowledge of their Associated operations. L4
5. Analyze the importance of algebraic properties with regard to working with in various number systems. L4

UNIT I
Mathematical Logic
Statements and Notations, Connectives, Well Formed Formulas, Truth Tables, Tautology, Equivalenceimplication, Normal forms, Quantifiers, Universal quantifiers.

UNIT II
Predicates
Predicative logic, free and bounded variables, rules of inference, consistency, proof of contradiction.

UNIT III
Principles of Mathematical Induction
The well ordering principle, recursive definition, division algorithm, prime numbers, greatest commondivisor, Euclidean algorithm, fundamental theorem of arithmetic.

UNIT IV
Relations
Properties of Binary relations, equivalence, transitive closure, compatibility and partial ordering relations, Hasse diagram.
Functions: Inverse function, composition of functions, recursive functions.

UNIT V
Groups
Algebraic structures, examples and general properties, Semi groups and monoids, Groups and Sub groups, cosets and Lagranges theorem, homomorphism, and isomorphism of groups, cyclic groups, permutation groups.

Text Books:

1. Discrete Mathematics for Computer scientists & Mathematicians, J. L. Mott, A. Kandel, T.P.Baker.
2. Discrete mathematics and its Applications, Kenneth H.Rosen, fifth edition. TMH

Reference Books:

1. Elements of Discrete mathematics, C.L.Liu, D.P.Mohapatra, 4th edition, McGraw Hill education (India)Private Limited.
2. Discrete mathematical structures theory and applications- malik & Sen Cengage.
3. Discrete mathematics with applications, Thomas Koshy, Elsevier.
4. Logic and Discrete mathematics, grass Man & Trembley, Pearson Education.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
3/0/0 3

LOGIC CIRCUITS DESIGN (CESLC1)

Course Objective:
Understand the significance of converting from mechanical era to electronic era and fundamentals of assembly language.

Course Outcomes:
Upon completion of the course, the student will be able to:

1. Analyze number systems, perform base conversions, and apply Boolean algebra to implement digital logic gates.(Analysis) L4
2. Apply the K-Map method to minimize logic expressions, design combinational circuits for various applications,and solve problems related to code converters, arithmetic operations, and datamanipulation. (Application and Analysis) L4
3. Evaluate flip-flops, design and analyze clocked sequential circuits, and create
4. registers, shift registers, and counters for digital systems.(Evaluation and Synthesis) L5
   Examine register transfer language, performs register transfers, and applies micro-operations for arithmetic,logic, and shifting operations in digital systems. (Analysis and Application) L4
5. Explore memory organization, decoding techniques, and programmable logic devices to understand Memory systems in digital systems.(Analysis) L4

UNIT I

Number System & Logic Gates
Digital Systems, Binary Numbers, Number base conversions, Octal, Hexadecimal and other base numbers, Excess-3 code, Gray code, complements, signed binary numbers, binary codes, binary storage and registers, binary logic, Boolean algebra and logic gates, Basic theorems and properties of Boolean Algebra, Boolean functions, canonical and standard forms, Digital Logic Gates, Implementation of basic gates using universal gates.

UNIT II

Gate -Level Minimization
The K-Map Method (3, 4, 5, Variables) sum of products, product of sums simplification, don't care conditions.

Combinational Circuits (CC)
Analysis procedure, Design Procedure, Combinational circuit for different code converters and other problems, Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, De-multiplexers.

UNIT III

Flip-Flops
Basic Latch, SR and D latches, Master Slave edge triggered D Flip-flop, T Flip-Flop, and JK Flip Flops, Analysis of clocked sequential circuits. State Reduction and assignment, Flip-Flop Excitation tables, Design procedure. Registers, Shift registers, Ripple counters, Synchronous counters, other counters.

UNIT IV

Register Transfer and Micro-operations

Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro-operations, Shift Micro-operations, Arithmetic Logic Shift Unit.

UNIT V

Memory

Introduction, Random-Access memory, Memory decoding, ROM, Programmable Logic Array, Programmable Array Logic, Sequential Programmable Devices.

Text Books:

1. Digital Design, M. Morris Mano, M.D.Ciletti, 5th edition, Pearson.
2. Computer System Architecture, M.Morris Mano, 3rd edition, Pearson.

Reference Books:

1. Fundamentals of Logic Design, C. H. Roth, L. L. Kinney, 7th edition, Cengage Learning.
2. Fundamentals of Digital Logic & Micro Computer Design, 5TH Edition, M. Rafiquzzaman, John Wiley.
3. R.P.Jain “Modern Digital Electronics” Tata McGraw Hill, 4th edition, 2009.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
3/0/0 3

DATABASE MANAGEMENT SYSTEMS (C53PC1)

Course Objective:
To provide students an in depth understanding of the fundamentals of relational systems including data models, databases.

Course Outcomes:
Upon completion of the course, the student will be able to:

1. Develop a foundational understanding of database management systems(DBMS), their evolution, importance, and the fundamental concepts that govern their design and operation. L3
2. Compare and examine the expressive power, capabilities, advantages, and limitations of relational algebra and calculus in querying relational databases. L4
3. Develop expertise in formulating complex SQL queries, constraints, conditions, andexpressions to retrieve, manipulate, and manage data based on specific criteria, requirements, and business rules. L3
4. Develop a foundational understanding of transactions, their definition, properties, and significance in ensuring data consistency, integrity, and reliability in database systems. L3
5. Compare and contrast index data structures with traditional file organizations. L4

UNIT I
Introduction and Basic Concepts
File organization for conventional data management system, Higher-level file organization for DBMS, Database System Applications, Purpose of Database Systems, View of Data, Database Languages – DDL, DML, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database Users and Administrators. Relational Model-Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity, constraints, Querying relational data, Logical data base Design: ER to Relational, Introduction to Views, Destroying /Altering Tables and Views.

UNIT II
Relational Algebra and Calculus
Preliminaries, Relational Algebra, Relational calculus Tuple relational Calculus, Domain relational calculus,Expressive Power of Algebra and calculus.
Introduction to Database design: Database Design and ER diagrams, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises.

UNIT III
SQL
SQL data definition and Data types, Schema and catalog concepts in SQL, Queries, Constraints, Triggers: Form of Basic SQL Query, UNION, INTERSECT, and EXCEPT, Nested Queries, Aggregate Operators, NULL values, Complex Integrity Constraints in SQL, Triggers and Active Data bases, Designing Active Databases, NoSQL database (MongoDB introduction).
Schema Refinement and Normal Forms: Introduction to Schema Refinement, Functional

Dependencies - Reasoning about FDs, Normal Forms - 1NF, 2NF, 3NF, Properties of Decompositions, Normalization, Schema Refinement in Database Design, Other Kinds of Dependencies.

UNIT IV

Transaction Management

Transactions, Transaction Concept, A Simple Transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, serializability and Atomicity, Transaction Isolation Levels, Implementation of

Isolation Levels. Concurrency Control, Lock–Based Protocols, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols, Multi-version Schemes. Recovery System-Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with loss of non-volatile storage, Early Lock Release and Logical Undo Operations, Remote Backup systems.

UNIT V

Indexing

Index Data Structures, and Comparison with File Organizations. Tree-Structured Indexing, Intuition for tree Indexes, Indexed Sequential Access Method (ISAM), B+ Trees: A Dynamic Index Structure, Search, Insert, Delete., Hash- Based Indexing, Static Hashing, Extendible hashing, Linear Hashing, Extendible vs. Linear Hashing.

Text Books:

1. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, McGraw Hill Education(India) Private Limited, 3rd Edition.
2. Database System Concepts, A. Silberschatz, Henry. F. Korth, S. Sudarshan, McGraw Hill Education(India) Private Limited l, 6th edition.

Reference Books:

1. Database Systems, 6th edition, R Elmasri, Shamkant B.Navathe, Pearson Education.
2. Database System Concepts, Peter Rob & Carlos Coronel, Cengage Learning.
3. Introduction to Database Management, M. L. Gillenson and others, Wiley, Student Edition.
4. Database Development and Management, Lee Chao, Auerbach publications, Taylor & Francis Group.
5. Introduction to Database Systems, C. J. Date, Pearson Education.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
3/0/0 3

DATA STRUCTURES (C53PC2)

Course Objective:
Make to understand the significance of data structures and imply them in building efficient algorithms.

Course Outcomes:

1. Use asymptotic notations to calculate the time and space complexity of a particular algorithm and Differentiate linear and nonlinear data structures. L3
2. Identify appropriate data structure for a given problem scenario. Illustrate array and linked representations Of stack and queues. L3
3. Apply the knowledge of binary tree to implement tree traversal techniques, demonstrate how priority queues are implemented. Perform operations of max heap and disjoint set ADT with an example. L3
4. Compare searching, sorting and external sorting methods. L4
5. Use graph search methods and perform operations on various search trees. L3

UNIT I

Basic Concepts

Data objects and Structures, Algorithm Specification-Introduction, Recursive algorithms, Data abstraction, Performance analysis- Time complexity and Space complexity, Asymptotic Notation-Big O, Omega and Theta notations, Complexity Analysis Examples, Introduction to Linear and Non Linear data structures.

UNIT II

Representation of Single, Two Dimensional Arrays and their Applications

Sparse matrices-array and linked representations. Linear list ADT-array representation and linked list representation, Singly Linked Lists-Operations Insertion, Deletion, Circular linked lists-Operations for Circular linked lists, Doubly Linked Lists Operations- Insertion, Deletion. Stack ADT, definition, array and linked list implementations, applications-infix to postfix conversion, Postfix expression evaluation, recursion implementation, Queue ADT, definition, array and linked list, Implementations, Circular queues- Insertion and deletion operations, Polynomial.

UNIT III

Trees

Definition, terminology, Binary trees-definition, Properties of Binary Trees, Binary Tree ADT, representation of Binary Trees-array and linked representations, Binary Tree traversals, threaded binary trees, Priority Queues –Definition and applications, Max Priority Queue ADT implementation-Max Heap- Definition, Insertion into a Max Heap, Deletion from a Max Heap.
Disjoint set ADT - Equivalence relations, the dynamic equivalence problem, Basic data structure, Smart union algorithms, Path compression, worst case for union by rank and path compression, and an application - generation of mazes.

UNIT IV

Searching

Linear Search, Binary Search, Hashing-Introduction, hash tables, hash functions, Overflow

Handling,Comparison of Searching methods.
Sorting: Insertion Sort, Selection Sort, Radix Sort, Quick sort, Heap Sort, Merge sort, Comparison ofSorting methods.External sorting-Model for external sorting, basic external sorting algorithm, multi-way merge, poly- phasemerge, replacement selection.

UNIT V

Graphs

Definitions, Terminology, Applications and more definitions, Properties, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph Search methods - DFS and BFS, Complexity analysis. Search Trees Binary Search Tree ADT, Definition, Operations- Searching, Insertion and Deletion, Balanced search trees-AVL Trees-Definition and Examples only, B-Trees-Definition and Examples only, Red Black Trees-Definitions and Examples only, k-d trees, Comparison of Search Trees.

Text Books:

1. Data structures, Algorithms and Applications in C++, 2nd Edition, Sartaj Sahni, Universities Press.
2. Data structures and Algorithms in C++, Adam Drozdek, 4th edition, Cengage learning.

Reference Books:

1. Data structures with C++, J. Hubbard, Schaum’s outlines, TMH.
2. Data structures and Algorithms in C++, M.T. Goodrich, R. Tamassia and D. Mount, Wiley India.
3. Data structures and Algorithm Analysis in C++, 3rd edition, M. A. Weiss, Pearson.
4. Classic Data Structures, D. Samanta, 2nd edition, PHI.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
3/0/0 3

OPERATING SYSTEMS (C53PC3)

Course Objective:
To provide students an in depth understanding of basic components of a computer operating system, scheduling policies, deadlocks, memory management, synchronization, system calls and file systems.

Course Outcomes:
Upon completion of the course the student will be able to:

1. Identify the components and structure of Operating System and the services provided by an Operating System and Analyze the role of Operating System to develop software applications. L4
2. Identify and solve issues related to the critical section problem by using the knowledge of process scheduling and implement the existing algorithms to evaluate the performance. L3
3. Analyze the process scheduling to handle and prevent deadlocks, compare and contrast memory management strategies for efficient utilization of memory. L4
4. Analyze the performance of demand paging and page replacement algorithms, and mplementation of system calls for a given file system. L4
5. Analyze the role of mass storage in supporting file systems and protection mechanisms in preventing unauthorized access. L4

UNIT I
Overview
Introduction-Operating system objectives, User view, System view, Operating system definition, Computer System Architecture, OS Structure, OS Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments. Operating System services, User and OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, OS Structure.

UNIT II
CPU Scheduling Process Concepts
The Process, Process State, Process Control Block, Threads, Process Scheduling, Scheduling Queues, Schedulers, Context Switch, Operations on Processes, System calls fork(),exec(),wait(),exit(), Inter-process communication-ordinary pipes and named pipes, message queues, shared memory, in Unix.
Process Scheduling-Basic concepts, Scheduling Criteria, Scheduling algorithms, Multiple-Processor Scheduling, Real Time Scheduling, Thread scheduling, Linux scheduling and Windows scheduling. Process Synchronization, Background, The Critical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization in Linux and Windows.

UNIT III
Deadlocks-System
Deadlocks-System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.
Memory Management: Memory Management Strategies, Background, Swapping, Contiguous

Memory Allocation, Segmentation, Paging, Structure of Page Table.

UNIT IV

Virtual Memory

Virtual Memory Management Background, Demand Paging, Copy-on-Write, Page Replacement, Page Replacement Algorithms, Allocation of Frames, Thrashing, Virtual memory in Windows. Storage Management File System- Concept of a File, System calls for file operations - open (), read(), write (), close (), seek (), unlink (), Access methods, Directory and Disk Structure, File System Mounting, File Sharing.

UNIT V

File System

File System Structure, File System Implementation, Directory Implementation, Allocation methods, Free- space Management, Efficiency, and Performance. Overview of Mass Storage Structure. System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

Text Book:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, 8th Edition, Wiley,2016 India Edition.
2. Operating Systems – Internals and Design Principles, W. Stallings, 7th Edition, Pearson.

Reference Books:

1. Modern Operating Systems, Andrew S Tanenbaum, 3rd Edition, PHI
2. Operating Systems: A concept-based Approach, 2nd Edition, D.M. Dhamdhere, TMH.
3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
4. An Introduction to Operating Systems, P.C.P. Bhatt, PHI.
5. Principles of Operating systems, Naresh Chauhan, Oxford University Press.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
0/0/2 1

LOGIC CIRCUITS DESIGN LAB (CESCL2)

Course Objective:
Understand and implement the Logic gates, combinational and sequential Circuits using hardware components.

Course Outcomes:
Upon completion of the course the student will be able to:
1. Demonstrate adeptness in utilizing the basic tenets of digital logic gates to create effective designs for both combinational and sequential circuits, showcasing a sound understanding at. L3
2. Utilize advanced analytical skills to thoroughly examine and interpret results stemming from logic gates and a diverse range of combinational and sequential circuits, attaining an elevated level of understanding at. L4

Exercises in Digital Logic Design:
1. Design Logic gates using minimum number of Universal (NAND and NOR)gates.
2. Identify the logic gates required to design a Full Adder, and Design it using them.
3. Analyze and formulate the relationship between input lines and control lines in Multiplexers.
4. Design and implement the 4:1 MUX, 8:1 MUX using gates/ICs.
5. Design and Implement a 3 to 8 decoder using gates.
6. Design a 4 bit comparator using gates/IC.
7. Design and Implement a 4 bit shift register using Flip flops and draw the timing diagrams.
8. Design and Implement a Decade counter and draw the timing diagram.

i. Deign a 4-bit Gray to Binary and Binary to Gray Converter.
ii. Design a 16 bit Adder/ Subtractor using 4-bit Adder /Subtractor IC's.
iii. Design a 3x8 Decoder.
iv. Design a16x4 priority encoder using two 8x3 priority encoder.
v. Design a 16x1 multiplexer using 8x1 multiplexer.
vi. Design a 16bit comparator using 4 bit comparators.
vii. Design an 8 bit parallel load and serial out shift register using two 4 bit shift register.
viii. Design an 8 bit serial in and serial out shift register using two 4 bit shift register.
ix. Deign a Ring counter and twisted ring counter using a 4-bit shift register.
x. Design a model to 53 counter using two decade counters.
xi. Design a 4 digit hex counter using synchronous one digit hex counters.
xii. Design a 4 digit hex counter using Asynchronous one digit hex counters.

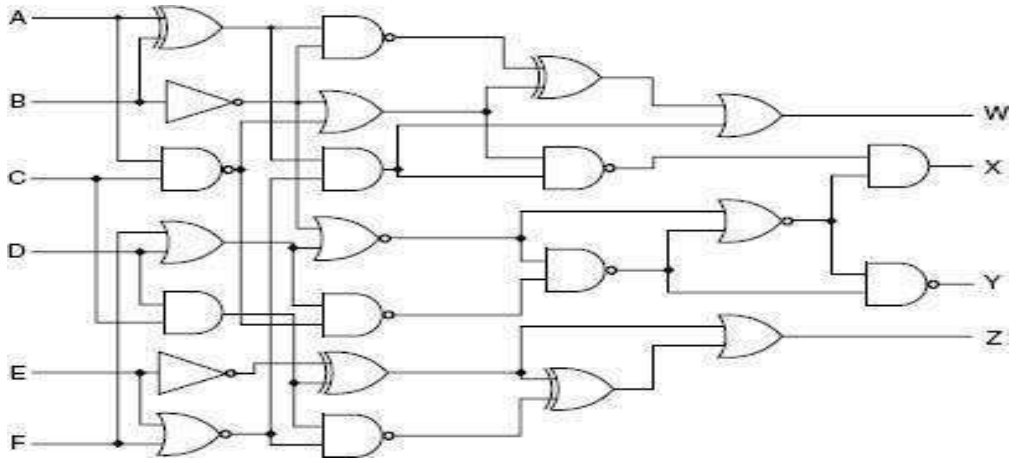
9. Design a 4 bit pseudo random sequence generator using 4-bit ring counter.

Experiment # 01
Realization of Combinational circuits

Description: Five binary inputs of a digital logic circuit are designated as A, B, C, D and E. The circuit has three outputs, namely X, Y and Z. X should output 0, only if A is 0 and C and E both are 1. In all other cases X must remain 1. Y should output 0, if both B and C are 0 and D and E are 1. In all other cases Y must remain as 1. Z goes low if A, D and E are 1 and B and C are 0. Otherwise Z remains high. Prepare a truth table for the logic and then design a suitable circuit to implement it.

Task1: Identify the Logic gates required with their functionalities to implement the given scenario
Task2: Apply different inputs to the Logic gates and check the functionality
Task3: Prepare a Complete Truth table for the scenario
Task4: Design a suitable circuit to implement it.

1. a) Generate the truth table of the following circuit with six inputs A, B, C, D, E and F and fouroutputs W, X, Y and Z.



1) Study the Truth tables for different Logicgates
2) Apply different inputs to the Logic gates and check thefunctionality
3) Analyse the outputs of the Logicgates
4) Generate the truth table for theCircuit.

A circuit has five inputs as A, B, C, D and E. Its six outputs are U, V, W, X, Y and Z. Design thecircuit defined by the following truth table.

A	B	C	D	E	U	V	W	X	Y	Z
1	0	1	X	x	0	1	x	x	x	x
X	x	0	1	0	x	x	1	1	x	x
1	x	X	X	1	1	x	x	x	0	1
X	1	X	0	x	x	0	0	0	1	0

1) Study and Analyze different input combinations for the given table.
2) Develop the Truth table from the given table.
3) Design the circuit from the truth table.
4) Verifytheoutput.

Design a 4-bit combinational circuit to increment (A circuit that adds one to a 4-bit binary number)using 4 Half-adders.

1) Understand the combinational circuit design
2) Analyze the functionality of an increment circuit.
3) Design the circuit using half adders

Experiment # 02
Design of ALU with two select-lines

Description: Design a 4-bit ALU with inputs A0–A3, B0–B3 and Cin. It is to have 4-bit output Y0–Y3 and Cout. The ALU would have two select lines to implement any one of the following four functions as per the pattern of select lines.

Select lines	Function	Remarks
00	A + B + Cin	Add with carry
01	A-B-Cin	Subtract with borrow
10	A AND B	Logical AND
11	A OR B	Logical OR

Task1: Identify the Logic gates required to design an ALU.
Task2: Design the Adder / subtractor circuit of ALU with the given condition. **Task3:** Design the Logical AND/Logical OR circuit of ALU with the given condition. **Task4:** Test the circuit to get the required output.

Experiment # 03
Design of ALU with two states

Description: Design an ALU capable of performing multiplication of two 4-bit numbers by producing 8-bit result,

using Booth's algorithm. The unit may be hardware controlled in which case the complete hardware design to be implemented. Alternately the unit may micro-coded, in which case all signals and micro-steps are to be specified.
Task1: Identify the Logic gates required to design an ALU
Task2: Design the ALU circuitry for Multiplication
Task3: Test the circuit with different inputs for multiplication
Task4: Verify and record the output.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
0/0/2 1

DATABASE MANAGEMENT SYSTEMS LAB (C53PC4)

Course Objective:
Emphasize on designing, developing and querying a database in the context of example database "Roadwaytravels".

Course Outcomes:
Upon completion of the course the student will be able to:

1. Design and implement a database schema for a given problem. L5
2. Apply the normalization techniques for development of application software to realistic problems. L3
3. Formulate queries using SQL DML/DDL/DCL commands. L5
4. Develop application programs using PL/SQL. L5

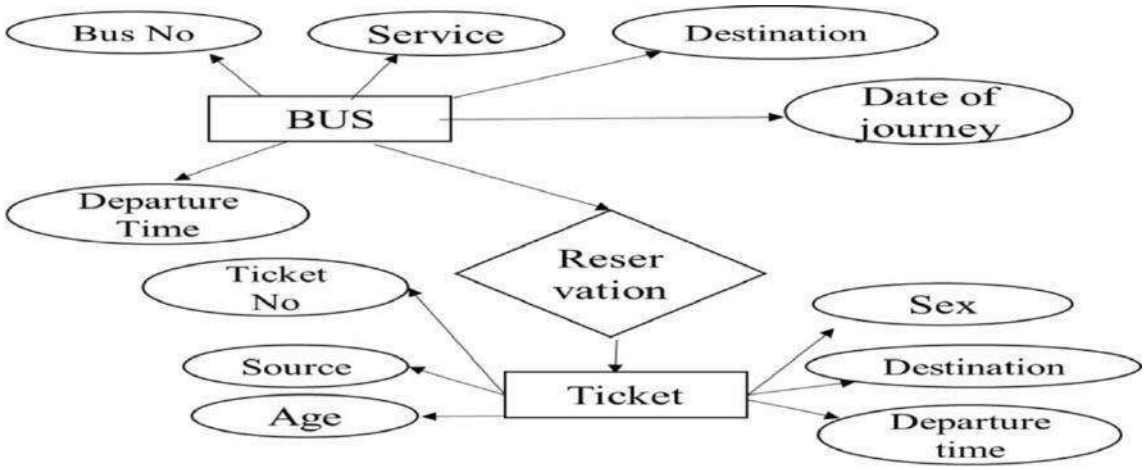
Roadway Travels
"Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad. The company wants to computerize its operations in the followingareas:
Reservation & Cancellation
Reservations are directly handled by booking office, Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family). Cancellations are also directly handed at the booking office.
In the process of computerization of Roadway Travels you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop queries using SQL to retrieve the data from the database.
The above process involves many steps like
1. Analyzing the problem and identifying the Entities andRelationships,
2. E-RModel
3. Relational Model
4. Normalization
5. Creating the database
6. Querying Students are supposed to work on these steps week wise and finally create a complete "Database System" to Roadway Travels. Examples are given at every experiment for guidance to students.

Experiment 1: E-R Model
Analyse the problem carefully and come up with the entities in it. Identify what data has to be persisted inthe database. This contains the entities, attributes etc.
Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.Example:
Entities: 1. Bus 2. Ticket 3. Passenger
Relationships 1. Reservation. 2. Cancellation.
Primary Key Attributes: 1. Ticket ID (Ticket Entity) 2. Passport ID (Passenger Entity)
3. Bus_No. (Bus Entity)
Apart from the above mentioned entities you can identify more. The above mentioned are few.

Experiment 2: Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

Example: E-R diagram for bus



Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi- valued, and Derived) have different way of representation.

Example: The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

Passenger Name	Age	Sex	Address	Passport ID	Ticket_ ID

Experiment 4: Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data with in the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only. For the above table in the First normalization we can remove the multi valued attribute Ticket id and place it in another table alongwith the primary key of passenger.

First Normal Form: The above table can be divided into two tables as shown below.

Passenger Name	Age	Sex	Address	Passport ID

Passpo rtID	Ticket _ID

You can do the second and third normal forms if required. Any, how Normalized tables are given at the end.

Experiment 5: Installation of Mysql and practicing DDL commands

Installation of MySql. In this week student will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. Student will also try truncate, rename commandsetc.

Example for creation of a normalized "Passenger" table.

CREATE TABLE Passenger (Passport_id INTEGER PRIMARY KEY, Name VARCHAR (50) Not NULL, Age Integer Not NULL, Sex Char, Address VARCHAR (50) Not NULL); Similarly createall other tables.

Experiment 6: Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

SELECT - retrieve data from the a database INSERT - insert data into a tableUPDATE - updates existing data within a table

DELETE - deletes all records from a table, the space for the records remain Inserting values into "Bus" table:

Insert into Bus values (1234,'hyderabad', 'tirupathi'); Insert into Bus values (2345,'hyderabd','Banglore'); Insert into Bus values (23,'hyderabd','Kolkata'); Insert into Bus values (45,'Tirupathi,'Banglore'); Insert into Bus values (34,'hyderabd','Chennai'); Inserting values into "Passenger" table:

Insert into Passenger values (1, 45,'ramesh', 45,'M','abc123'); Insert into Passenger values (2, 78,'geetha', 36,'F','abc124'); Insert into Passenger values (45, 90,'ram', 30,'M','abc12'); Insert into Passenger values (67, 89,'ravi', 50,'M','abc14'); Insert into Passenger values (56, 22,'seetha', 32,'F','abc55'); Few more Examples of DML commands:

Select * from Bus; (selects all the attributes and display)UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;

Experiment 7: Querying

In this week student are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Practice the following Queries

1. Display unique PNR_No of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with 'r' and ends with'h'.
5. Find the names of passengers whose age is between 30 and45.
6. Display all the passengers names beginning with'A'
7. Display the sorted list of passengers names

Experiment 8: and Experiment 9: Querying (continued…)

Student are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX andMIN), GROUP BY, HAVING and Creation and dropping of Views.

Write a Query to display the Information present in the Passenger and cancellation tables.

1. Display the number of days in a week on which the 9W01 bus is available.
2. Find number of tickets booked for each PNR_no using GROUP BYCLAUSE.
3. Find the distinct PNR numbers that are present.

4. Find the number of tickets booked by a passenger where the number of seats is greater than1. Hint: Use GROUP BY, WHERE and HAVING CLAUSES.
5. Find the total number of cancelled seats.

Experiment 10: PL/SQL

1. Write a PL/SQL block for Addition of Two Numbers
2. Write a PL/SQL block for IF Condition
3. Write a PL/SQL block for IF and else condition
4. Write a PL/SQL block for greatest of three numbers using IF ANDELSEIF
5. Write a PL/SQL block for summation of odd numbers using for LOOP.

Experiment 11: Triggers

Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.
Eg: CREATE TRIGGER updcheck BEFORE UPDATE ON passenger
FOR EACH ROWBEGIN
IF NEW.TickentNO> 60 THEN SET New.Tickent no = Ticket no; ELSESET New.Ticketno = 0; END
IF; END;

Experiment 12: Procedures

Learn creation of stored procedure, Execution of procedure and modification of procedure. Practiceprocedures using the above database.
Eg: CREATE PROCEDURE myProc() BEGIN
SELECT COUNT (Tickets) FROM Ticket WHERE age>=40; End;Experiment 13 Cursors
Declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local
variables as needed from the cursor, one row at a time. Close the cursor when done. CREATE PROCEDURE myProc(in_customer_id INT) BEGIN
DECLARE v_idINT;
DECLARE v_name VARCHAR (30);
DECLARE c1 CURSOR FOR SELECT stdId, stdFirstname FROMstudents WHEREstdId=in_customer_id;
OPEN c1;
FETCH c1 into v_id, v_name; Close c1; END;Tables BUS
Bus No: Varchar: PK (public key) Source : Varchar Destination : VarcharPassenger
PPNO: Varchar (15)) : PK Name: Varchar (15) Age : int (4) Sex:Char (10) : Male / Female Address:VarChar (20)
Passenger_Tickets
PPNO: Varchar (15)) : PK Ticket_No: Numeric (9)Reservation
PNR_No:Numeric(9):FKJourney_date:datetime(8)No_of_seats:int(8)Address:Varchar(50)
Contact_No:Numeric (9) --> Should not be less than 9 and Should not accept any other character other than Integer Status: Char (2) : Yes /No
Cancellation
PNR_No: Numeric(9) : FK Journey_date : datetime(8) No_of_seats : int (8) Address : Varchar (50) Contact_No: Numeric (9) --> Should not be less than 9 and Should not accept any other character otherthan Integer Status: Char (2) : Yes / No
Ticket
Ticket_No: Numeric (9): PK Journey_date : datetime(8) Age : int (4) Sex:Char(10) : Male / Female Source :Varchar Destination : Varchar Dep_time : Varchar

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech III Semester **L/T/P C**
0/0/2 1

DATA STRUCTURES LAB (C53PC5)

Course Objective:
Write and execute programs in C++ to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables, search trees and implement various sorting and searching algorithms.

Course Outcomes:
1. Apply single linked list, doubly linked list, stack and double ended queue data structures for the given Problem scenario and demonstrate the insertion, deletion and display operations. L3
2. Apply binary search tree algorithms to the given problem scenario and demonstrate the in order, preorder and post order traversals. L3
3. Apply various sorting algorithms for the given problem scenario. L3
4. Apply binary search tree, hashing and extendible hashing techniques for the given problem scenario. L3

C++ Programming Concepts
Review of C, input and output in C++, functions in C++-value parameters, reference parameters, Parameter passing, function overloading, function templates, Exceptions- throwing an exception and handling an exception, arrays, pointers, new and delete operators, class and object, access specifies, friend functions, constructors and destructor, Operator overloading, class templates, Inheritance and Polymorphism.

List of Programs to be performed during the Course
1.Write a C++ program that uses functions to perform the following:
a) Create a singly linked list of integers.
b) Delete a given integer from the above linked list.
c) Display the contents of the above list after deletion.
2.Write a template based C++ program that uses functions to perform the following:
a) Create a doubly linked list of elements.
b) Delete a given element from the above doubly linked list.
c) Display the contents of the above list after deletion.
3. Write a C++ program that uses stack operations to convert a given infix expression into its postfix equivalent, Implement the stack using an array.
4.Write a C++ program to implement a double ended queue ADT using an array, using a doubly linked list.
5.Write a C++ program that uses functions to perform the following:
a) Create a binary search tree of characters.
b) Traverse the above Binary search tree recursively in preorder, in order and post order.
6.Write a C++ program that uses function templates to perform the following:
a) Search for a key element in a list of elements using linear search.
b) Search for a key element in a list of sorted elements using binary search.
7.Write a C++ program that implements Insertion sort algorithm to arrange a list of integers in ascending order.

8.Write a template based C++ program that implements selection sort algorithm to arrange a listof elements in descending order.

9.Write a template based C++ program that implements Quick sort algorithm to arrange a list ofelements. in ascending order

10.Write a C++ program that implements Heap sort algorithm for sorting a list of integers in ascending.
order

11.Write a C++ program that implements Merge sort algorithm for sorting a list of integers in ascending
order.

12.Write a C++ program to implement all the functions of a dictionary (ADT) using hashing.

13.Write a C++ program that implements Radix sort algorithm for sorting a list of integers in ascending
order.

Write a C++ program that uses functions to perform the following:

1.Create a binary search tree of integers.

2.Traverse the above Binary search tree non-recursively in ignored.

3.Write a C++ program that uses functions to perform the following:

4.Create a binary search tree of integers.

5.Search for an integer key in the above binary search tree non-recursively.

6.Search for an integer key in the above binary search tree recursively.

7.Write a C++ program to implement hashing using any hash function.

8.Write a C++ program to implement extendible hashing.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
3/0/0 3

PROBABILITY & STATISTICS (CBSM3)

Course Objectives:
To learn:
1. Random variables that describe randomness or an uncertainty in certain realistic situation.
2. The study of discrete and continuous distribution predominantly describes important probabilitydistribution.
3. Sampling distribution of mean, variance, point estimation and interval estimation.
4. The testing of Hypothesis of Large samples.
5. The testing of Hypothesis of small samples.
6. The basic ideas of statistics including correlation and regression.

Course Outcomes:
After learning the contents of this course, the student must be able to learn the concept of
1. Random variables and various discrete and continuous probability distributions and their properties to solve problems. L3
2. Illustrate interval estimations of Mean and Proportion of large samples. L3
3. Analyze important decisions for few samples which are taken from a large data. L4
4. Test the hypothesis and give the inference to the given data. L4
5. Apply statistical methods of studying data sample. L3

UNIT I
Random Variables & Distributions
Random variables –Discrete and continuous, Mass Function, Density function of probability distributions, Binomial, Poisson and Normal distributions related properties.

UNIT II
Sampling Distributions
Sampling distributions of means (σknown and unknown).Estimation theory, point estimations, Interval estimations, Maximum Error.

UNIT III
Large Samples
Null hypothesis, alternative hypothesis, Type-1, Type-2 errors, Critical region, confidence interval for mean, testing of single mean and two means, confidence interval for the proportions ,Test of Hypothesis for the single proportions and difference between the proportions.

UNIT IV
Small Samples
Small samples Test for single mean, difference of two means, test for ratio of variances, Chi-square test forgoodness of fit and independence of attributes.

UNIT V
Basic Statistics
Correlation and regression, Rank correlation, Curve fitting by the method of least squares, fitting of straightlines, second degree parabolas, power and exponential curves.

Text Books:

1. Probability & Statistics for Engineers by G.S.S. BhismaRao, SciTech Publications.
2. Probability & Statistics for Engineers by D.K.Murugesan & P.Guru Swamy, Anuradha Publications.

Reference Books:

1. W.Feller- An introduction to probability theory and its applications- Vol.1- 3rd edition Wiley-1968.
2. Probability & Statistics for Engineers, Millers and John E.Freund, Prentice Hall of Ind.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
3/0/0 3

COMPUTER ORGANIZATION AND ARCHITECTURE (C54PC1)

Course Objective:
Understand the detailed computer architecture and organization, hardware operation of digital computer.

Course Outcomes:
Upon completion of the course the student will be able to:

1. Make use of the principles of computer organization of a computer. L3
2. Create designs for both hardwired and micro-programmed control units, showcasing proficiency in the architectural aspects of 8086. L3
3. Evaluate computer arithmetic operations and develop fundamental Assembly Language Programs for the 8086 processor. L4
4. Assess I/O data transfer modes and memory hierarchy in detail. L4
5. Analyze how the principles and mechanisms of concurrent processing are useful for processing computer systems. L4

UNIT I
Digital Computers
Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture. Basic Computer Organization and Design Instruction codes, Computer Registers, Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt, Complete Computer Description.

UNIT II
Central Processing Unit
Processor Organization, Register Organization, Instruction cycle, hardwired control unit, Micro program control unit. The 8086 Processor Architecture, Register organization, Physical memory organization, General Bus Operation, I/O Addressing Capability, Special Processor Activities, Minimum and Maximum mode system and timings.

UNIT III
Computer Arithmetic
Introduction, The arithmetic logic unit, Integer representation, Integer arithmetic, Floating point representation, Floating point arithmetic, Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating - point Arithmetic operations. Data Transfer and Manipulation Instructions and 8086 ALP basic programs.

UNIT IV
Input-Output Organization
Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, PriorityInterrupt, Direct memory Access, Input –Output Processor (IOP),Intel 8089 IOP.
Memory Organization Memory Hierarchy, Auxiliary memory, Associate Memory, Cache Memory, VirtualMemory, Memory Management Hardware.

UNIT V

Pipeline and Vector Processing

Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, CISC,RISC versusCISC, Vector Processing, Array Processors.

Multi Processors

Characteristics of Multiprocessors, Interconnection Structures, Inter processor arbitration, Inter processor communication, and synchronization.

Text Books:

1. Computer System Architecture, M. Moris Mano, Third Edition, Pearson.
2. Advanced Microprocessors and Peripherals, K M Bhurchandi, A.K Ray, 3rd edition, McGrawHillIndia Education Private Ltd.

Reference Books:

1. Microprocessors and Interfacing, D V Hall, SSSP Rao, 3rd edition, McGraw Hill India Education Private Ltd.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th Edition,Tata McGrawHill, 2002.
3. Computer Organization and Architecture, William Stallings, 9th Edition, Pearson.
4. David A. Patterson, John L. Hennessy: Computer Organization and Design – The Hardware SoftwareInterface ARM Edition, 4th Edition, Elsevier, 2009.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
3/0/0 3

SOFTWARE ENGINEERING (C54PC2)

Course Objective:
Apply the software engineering lifecycle by demonstrating competence in communication, planning,analysis, design, construction and deployment

Course Outcomes:
Upon completion of the course the student will be able to:

1. Identify common patterns in software development processes and apply different process models for software engineering. L3
2. Classify various requirements and develop techniques for gathering and analyzing requirements. L4
3. Apply existing fundamental design concepts in designing a software architecture based on given requirements. L3
4. Implement various testing methods to assess the overall quality of software. L4
5. Identify potential risks associated with software development. L3

UNIT I
Introduction to Software Engineering
The evolving role of software, Changing Nature of Software, legacy software, software myths.
Generic View of Process Software engineering- A layered technology, a process framework, the Capability Maturity Model Integration (CMMI), process patterns, process assessment, personal and team processmodels.
Process Models
The Waterfall Model, Incremental Process Model, Evolutionary Process models, specialized process models, unified process.

UNIT II
Software Requirements
Functional and Non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. Requirements Engineering Process Feasibility studies, Requirements elicitation and analysis, requirements validation, requirements management.
System Models-Context Models, Behavioral Models, Data Models, Object Models, structured methods.

UNIT III
Design Engineering
Design Process and Design quality, Design concepts, the design model, pattern based software Design, Creating an Architectural Design, Software architecture, Data Design, Architectural styles and patterns. Architectural Design, assessing alternative architectural designs, mapping data flow into software architecture. Modeling Component-Level Design, Design class-based components, conducting component- level design, object constraint language, design conventional components.

UNIT IV
Testing Strategies
A strategic approach to software testing, testing strategies for conventional software, Black-Box

and White- Box testing. Validation testing, system testing, the art of debugging.Product Metrics Software Quality, Frame work for Product metrics, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance. Metrics for Process and Products Software Measurement, Metrics for Software Quality.

UNIT V

Risk Management

Reactive versus Proactive, Risk strategies, software risks, Risk identification, Risk projection, Risk refinement. RMMM, RMMM plan.

Text Books:

1. Software engineering A Practitioner's approach, Roger S Pressman, Sixth Edition McGrawHill International Edition.
2. Software Engineering: Ian Sommer ville, Seventh Edition, Pearson Education.

Reference Books:

1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India 2010.
2. Software Engineering: A Primer, Waman S. Jawadekar, Tata McGraw Hill, 2008.
3. Software Engineering Foundations, Yingxu Wang, Auerbach Publications 2008.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
3/0/0 3

DESIGN AND ANALYSIS OF ALGORITHMS (C54PC3)

Course Objective:
Understand the design paradigms for developing an algorithm and analyzing it for a given problem.

Course Outcomes:
Upon completion of the course the student will be able to:

1. Analyze mathematical analysis methods to explore algorithm performance and apply the divide-and-conquer technique for resolving computing problems. L4
2. Demonstrate disjoint set operations and implement the backtracking technique to address computing problems. L3
3. Implement the Greedy method to solve diverse computing problems. L3
4. Develop effective algorithms for typical engineering design scenarios through
a. the application of the dynamic programming technique. L5
5. Solve intricate problems by applying the branch-and-bound technique, and
a. Analyze the characteristics of NP- hard and NP-complete problems. L4

UNIT I
Introduction
Algorithm definition, Algorithm Specification, Performance Analysis-Space complexity, Time complexity, Randomized Algorithms. Divide and conquer- General method, applications – Binary search, Merge sort, Quick sort, Strassen's Matrix Multiplication.

UNIT II
Disjoint Set Operations
Disjoint set operations, union and find algorithms, AND/OR graphs, Connected Components and Spanning trees, Bi-connected components Backtracking-General method, applications the 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT III
Greedy Method
General method, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT IV
Dynamic Programming
General Method, applications- Chained matrix multiplication, All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Travelling sales person problem.

UNIT V
Branch and Bound
General Method, applications-0/1 Knapsack problem, LC Branch and Bound solution, FIFO Branch and Bound solution, Traveling sales person problem. NP-Hard and NP Complete problems-Basic concepts, Non-deterministic algorithms, NP – Hard and NP-Complete classes, Cook's theorem.

Text Book:

1. Fundamentals of Computer Algorithms, 2nd Edition,Ellis Horowitz, Sartaj Sahni and S.Rajasekharan,Universities Press.
2. Design and Analysis of Algorithms, P. H. Dave, H. B. Dave, 2nd edition, Pearson Education.

Reference Books:

1. Algorithm Design: Foundations, Analysis and Internet examples, M. T. Goodrich and R. Tomassia,John Wiley and sons.
2. Design and Analysis of Algorithms, S. Sridhar, Oxford Univ. Press
3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson Education.
4. Foundations of Algorithms, R. Neapolitan and K. Naimipour, 4th edition, Jones and Bartlett Studentedition.
5. Introduction to Algorithms, 3rd Edition, T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, PHI.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
3/0/0 3

FORMAL LANGUAGES & AUTOMATA THEORY (C54PC4)

Course Objective:
Understand the central ideas of theoretical computer science from the perspective of formal languages.

Course Outcomes:
Upon completion of the course the student will be able to:

1. Understand the fundamental concepts of Automata Theory, including alphabets, strings, languages, and problems, and apply this understanding to explore Deterministic Finite Automata(DFA) and Nondeterministic Finite Automata(NFA), with practical use in text search. L4
2. Apply regular expressions and algebraic laws to solve language recognition problems, and demonstrate the conversion between Finite Automata and Regular Expressions. L3
3. Design context-free grammars, analyze derivations, and identify and resolve ambiguity in grammars and languages. L4
4. Analyze context-free grammars, implement Chomsky Normal Form conversions, andcompute the intricacies in transforming grammars and pushdown automata. L4
5. Explore undecidability, use concepts in recursive languages and Turing machines, and understand the impact of computational limits in theoretical computer science. L4

UNIT I
Introduction
Introduction to Finite Automata, Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.
Deterministic Finite Automata, Nondeterministic Finite Automata, an application: Text Search, Finite Automata with Epsilon-Transitions, Finite automata with output Epsilon Mealy and Moore machines, Equivalence of Mealy and Moore machines.

UNIT II
Regular Expressions
Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Automata and Regular expressions, Converting DFA's to Regular Expressions, Converting Regular Expressions to DFA, Properties of Regular Languages-Pumping Lemma for Regular Languages, Applications of the Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT III
Context-Free Grammars
Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.
Push Down Automata:
Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and

CFG's, Deterministic Pushdown Automata, non-deterministic pushdown automata, power of DeterministicPushdown Automata and Non-Deterministic Pushdown Automata.

UNIT IV

Normal Forms for Context- Free Grammars

The Pumping Lemma for Context-Free Languages, Closure Properties of Context-Free Languages. Decision Properties of CFL's - Complexity of Converting among CFG's and PDA's, Running time of conversions to Chomsky Normal Form. Introduction to Turing Machines: Problems That Computers Cannot Solve, The Turing Machine, Programming Techniques for Turing Machines, Extensions to the basic Turing machine, Restricted Turing Machines, Turing Machines, and Computers.

UNIT V

Undesirability

A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Post's Correspondence Problem, Other Undecidable Problems, Intractable Problems: Polynomial time and space, Some NP-complete problems, The Classes P and NP, NP-Complete Problem.

Text Book:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E.Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.

Reference Books:

1. Introduction to Languages and the Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R,Pearson.
5. Theory of Computer Science – Automata languages and computation, Mishra and

a. Chandrashekaran,2nd edition, PHI.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
3/0/0 3

OBJECT ORIENTED PROGRAMMING THROUGH JAVA (C54PC5)

Course Objective:
To provide students with a comprehensive understanding of the features of object oriented paradigm using JAVA programming

Course Outcomes:
Upon completion of the course the student will be able to:

1. Apply concepts of inheritance, polymorphism, and class hierarchies in Java, and analyze the structure and features of Java programs, including data types, variables, control statements, and methods. L3
2. Classify and differentiate between stream-based I/O mechanisms, and analyze various I/O operations including file handling, serialization, and advanced features like enumerations, auto-boxing, and generics. L4
3. Analyze different exception types and handling mechanisms and evaluate Java thread model features including thread creation, synchronization, and inter-thread communication. L4
4. Classify and differentiate between various collection interfaces and classes, and apply collection algorithms and utility classes for effective data management. L4
5. Apply the Model-View-Controller (MVC) architecture, and implement various layout managers, event handling mechanisms, including event sources, listeners, and adapter classes in Java. L3

UNIT I
Object-Oriented Thinking
A way of viewing world – Agents and Communities, messages and methods, Responsibilities, Classes and Instances, Class Hierarchies- Inheritance, Method binding, Overriding and Exceptions, Summary of Object- Oriented concepts. History of object-oriented programming, overview of java, Object oriented design, Structure of java program, Java buzzwords, Data types, Variables and Arrays, operators, expressions, control statements, Introducing classes, Methods and Classes, String handling.
Inheritance: Inheritance concept, Inheritance basics, Member access, Constructors, Creating Multilevel hierarchy, super uses, using final with inheritance, Polymorphism-ad hoc polymorphism, pure polymorphism, method overriding, abstract classes, Object class, forms of inheritance-specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance.

UNIT II
Packages
Defining a Package, CLASSPATH, Access protection, importing packages.
Interfaces: Defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces.
Stream based I/O (java.io): The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, The Console class, Serialization, Enumerations, auto boxing, generics.

UNIT III

Exception Handling

Fundamentals of exception handling, Exception types, Termination or presumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes.

Multithreading , Differences between thread-based multitasking and process-based multitasking, Java threadmodel, creating threads, thread priorities, creating multiple threads, synchronizing threads, inter thread communication, dead lock, Suspending, resuming, and stopping threads.

UNIT IV

The Collections Framework (java.util)

Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hash table ,Properties, Stack, Vector, More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter.

UNIT V

GUI Programming

Introduction, limitations of AWT, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

Event Handling The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson

Reference Books:

1. An Introduction to programming and Object Oriented design using Java, J. Nino and F.A. Hosch, JohnWiley & sons.
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Object Oriented Programming through Java, P. Radha Krishna, and UniversitiesPress.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ.Press.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
0/0/2 1

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB (C54PC6)

Course Objective:
Implement object oriented concepts using java programming in real time applications.

Course Outcomes:
The student will be able to do:

1. Implement and analyze complex algorithms that leverage Java collections for Solving real-world problems. L4
2. Apply design principles to create abstract classes that model real-world entities with abstraction. L3
3. Utilize advanced features of the Java language, such as generics, lambda expressions, and streams, in the context of object-oriented design. L3
4. Identify and avoid common pitfalls leading to deadlocks in multithreaded programs. L3

List of Experiments to be performed during the Course

1. Use Eclipse or Net beans platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
4. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
5. Write a Java program for the following:

a) Create a doubly linked list of elements.
b) Delete a given element from the above list.
c) Display the contents of the list after deletion.

6. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.
7. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area ( ). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area ( ) that prints the area of the given shape.
8. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas.

9. Write a java program to display the table using Labels in Grid Layout.
10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (Hint: use hash tables).
12. Write a Java program that correctly implements the producer – consumer problem using the conceptof inter thread communication.
13. Write a Java program to list all the files in a directory including the files present in all its Subdirectories.
14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascendingorder.
15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and alsoshow the number of interchanges occurred for the given set of integers.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
0/0/2 1

COMPUTER ORGANIZATION & ARCHITECTURE LAB (C54PC7)

Course Objective:
To provide students with a practical knowledge on 8086 assembly level language programs

Course Outcomes:
After learning the contents of this course, the student will be able to:
1. Design algorithm and develop the assembly language program for different problems using 8086 Assembly Language Programming. L5

List of Experiments to be performed during the Course
Write assembly language programs for the following using GNU Assembler.
Write assembly language programs to evaluate the expressions:
1.a = b + c – d * e ii) z = x * y + w – v +u / k

a)Considering 8-bit, 16 bit and 32 bit binary numbers as b, c, d, e.

b) Considering 2 digit, 4 digit and 8 digit BCD numbers. Take the input inconsecutive memory locations and also Display the results by using "int xx" of 8086. Validate program for the boundary conditions.

2.Write an ALP of 8086 to take N numbers as input, and arrange in ascending and descending order.

3.Write an ALP of 8086 to take N numbers as input. Considering 8-bit, 16 bit binary numbers and 2 digit, 4digit and 8 digit BCD numbers.
a)Find max and minimum

b) Find average Display the results by using "int xx" of 8086 and validate the program for the boundary conditions.

4.Write an ALP of 8086 to take a string as input (in 'C' format)

a)Find the length

b) Find is it Palindrome or not

5.Write an ALP of 8086 to take a string as input (in 'C' format) find whether given stringis a substring or not.

6.Write an ALP of 8086 to take a string as input (in 'C' format)

a)Find the given number is an Armstrong number or not

b) Find the Fibonacci series for n numbers Display the results by using "int xx" of 8086.

7.Write an ALP to implement the above operations as procedure and call from the Main procedure.

8.Write an ALP of 8086 to find the factorial of a given number as a Procedure and call from the mainprogram to display the result.

TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(Sponsored by TKR Educational Society , Approved by AICTE, Affiliated by JNTUH, Accredited by NBA & NAAC with 'A' Grade)

COMPUTER SCIENCE & ENGINEERING

B.Tech IV Semester **L/T/P C**
0/0/2 1

DESIGN AND ANALYSIS OF ALGORITHMS LAB (C54PC8)

Course Objective:
To provide students with a practical knowledge on paradigms of problem solving.

Course Outcomes:
Upon completion of the course the student will be able to:

1. Demonstrating a deep understanding of algorithmic principles and the capability to innovate solutions to analyze graph algorithms. L4
2. Analyze and explain how graph structures are useful in implementing BFS and DFS methods, as well as in solving combinatorial problems like sum of subset and N Queen's problem through Backtracking. L4
3. Analyze various greedy techniques with Dijkstra's algorithm and demonstrate synthesis through Kruskal's and Prim's algorithms for Minimum Cost Spanning Trees. L4
4. Analyze the methods used to solve 0/1 Knapsack problem using Dynamic Programming approach and demonstrate the working of Floyd's algorithm with Open MP approach. L4
5. Analyze the phenomenon of the Traveling Salesperson problem through optimal and approximation algorithms. L4

List of Experiments to be Performed During the Course:

1.Obtain the Topological ordering of vertices in a given digraph.

2.Compute the transitive closure of a given directed graph using Warshall's algorithm.

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

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

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

6.Print all the nodes reachable from a given starting node in a digraph using BFS method.

7.Check whether a given graph is connected or not using DFS method

8.Find a subset of a given set S = {s1,s2,.....,sn} of n positive integer 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 MP and determine the speed-up achieved.

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