DATA COMMUNICATION SYSTEMS

Unit I:
INTRODUCTION TO DATA COMMUNICATIONS AND NETWORKING: Standards Organizations for Data Communications, Layered Network Architecture, Open Systems Interconnection, Data Communications Circuits, Serial and parallel Data Transmission, Data communications Circuit Arrangements, Data communications Networks, Alternate Protocol Suites.

SIGNALS, NOISE, MODULATION, AND DEMODULATION:

Unit II:
METALLIC CABLE TRANSMISSION MEDIA:

OPTICAL FIBER TRANSMISSION MEDIA:

Unit III:
DIGITAL TRANSMISSION:
Pulse Modulation, Pulse code Modulation, Dynamic Range, Signal Voltage –to-Quantization Noise Voltage Ration, Linear Versus Nonlinear PCM Codes, Compingding, PCM Line Speed, Delta Modulation PCM and Differential PCM.

MULTIPLEXING AND T CARRIERS:
Time- Division Multiplexing, T1 Digital Carrier System, North American Digital Multiplexing Hierarchy, Digital Line Encoding, T Carrier systems, European Time- Division Multiplexing, Statistical Time – Division Multiplexing, Frame Synchronization, Frequency-Division Multiplexing, Wavelength- Division Multiplexing, Synchronous Optical Network

Unit IV:
WIRELESS COMMUNICATIONS SYSTEMS:

Unit V:
TELEPHONE INSTRUMENTS AND SIGNALS:
The Subscriber Loop, Standard Telephone Set, Basic Telephone Call Procedures, Call Progress Tones and Signals, Cordless Telephones, Caller ID, Electronic Telephones, Paging systems.

THE TELEPHONE CIRCUIT:

Unit VI:
CELLULAR TELEPHONE SYSTEMS:

Unit VII:
DATA COMMUNICATIONS CODES, ERROR CONTROL, AND DATA FORMATS:
Data Communications Character Codes, Bar Codes, Error Control, Error Detection, Error Correction, Character Synchronization.

DATA COMMUNICATIONS EQUIPMENT:

Unit VIII:
DATA –LINK PROTOCOLS:

TEXT BOOKS:
1. Introduction to Data Communications and Networking, Wayne Tomasi, Pearson Education.

REFERENCE BOOKS
1. Data Communications and Networking, Behrouz A Forouzan, Fourth Edition. TMH.
II Year B.Tech. IT-I Sem

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

UNIT-I
Mathematical Logic: Statements and notations, Connectives, Well formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Quantifiers, universal quantifiers

UNIT-II

UNIT-III
Relations: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Lattices, Hasse diagram. Functions: Inverse Function Composition of functions, recursive Functions, Lattice and its Properties,

UNIT-IV
Algebraic structures: Algebraic systems Examples and general properties, Semi groups and monads, groups sub groups’ homomorphism, Isomorphism.

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

UNIT-VI
Recurrence Relation: Generating Functions, Function of Sequences Calculating Coefficient of generating function, Recurrence relations, Solving recurrence relation by substitution and Generating funds. Characteristics roots solution of In homogeneous Recurrence Relation.

UNIT-VII
Graph Theory: Representation of Graph, DFS, BFS, Spanning Trees, planar Graphs

UNIT-VIII
Graph Theory and Applications, Basic Concepts Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers

TEXT BOOKS:

REFERENCES:
4. Discrete Mathematics with Applications, Thomas Koshy, Elsevier
5. Logic and Discrete Mathematics, Grass Man & Trembley, Pearson Education.
Unit I:
C++ Class Overview- Basic OOP concepts, Class Definition, Objects, Class Members, Access Control, Class Scope, Constructors and destructors, parameter passing methods, Inline functions, static class members, this pointer, friend functions, dynamic memory allocation and deallocation (new and delete), exception handling.

Unit II:
Function Overloading, Operator Overloading, Generic Programming- Function and class templates, Inheritance basics, base and derived classes, inheritance types, base class access control, runtime polymorphism using virtual functions, abstract classes, streams I/O.

Unit III:
Algorithms, performance analysis- Time complexity and space complexity. Review of basic data structures- The list ADT, Stack ADT, Queue ADT, array and linked Implementations using template classes in C++. Trees – Basic Terminology, Binary tree ADT, array and linked representations, traversals, threaded binary trees.

Unit IV:
Dictionaries, linear list representation, skip list representation, operations insertion, deletion and searching, Hashing- hash table representation, hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing, comparison of hashing and skip lists.

Unit V:
Priority Queues – Definition, ADT, Realizing a Priority Queue using Heaps, Definition, insertion, Deletion, Heap sort, External Sorting- Model for external sorting, Multiway merge, Polyphase merge.

Unit VI:
Search Trees (Part1):
Binary Search Trees, Definition, ADT, Implementation, Operations- Searching, Insertion and Deletion, AVL Trees, Definition, Operations – Insertion and Searching

Unit VII:
Search Trees (part-II): B-Trees, Definition, B-Tree of order m, insertion, deletion and searching, Comparison of Search Trees
Graphs – Basic terminology, representations of Graphs, Graph search methods – DFS, BFS.

Unit VIII:
Text Processing - Pattern matching algorithms- Brute force, the Knuth-Morris-Pratt algorithm, Tries- Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

REFERENCES:
3. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
8. An Introduction to Data structures and Algorithms, J.A. Storer, Springer.
UNIT I


UNIT II
DIGITAL LOGIC CIRCUITS-I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions. Flip-flops.

UNIT III
DIGITAL LOGIC CIRCUITS-II: Registers, Shift Registers, Binary counters, Decoders, Multiplexers, Programmable Logic Devices.

UNIT IV
COMPUTER ARITHMETIC: Algorithms for fixed point and floating point addition, subtraction, multiplication and division operations. Hardware Implementation of arithmetic and logic operations, High performance arithmetic.

UNIT V
INSTRUCTION SET & ADDRESSING: Memory Locations and Addresses, Machine addresses and sequencing, Various Addressing Modes, Instruction Formats, Basic Machine Instructions. IA-32 Pentium example.

UNIT VI
PROCESSOR ORGANIZATION: Introduction to CPU, Register Transfers, Execution of Instructions, Multiple Bus Organization, Hardwired Control, Microprogrammed Control

UNIT VII
MEMORY ORGANIZATION: Concept of Memory, RAM, ROM memories, memory hierarchy, cache memories, virtual memory, secondary storage, memory management requirements.

UNIT VIII
INPUT / OUTPUT ORGANIZATION: Introduction to I/O, Interrupts- Hardware, Enabling and disabling Interrupts, Device Control, Direct memory access, buses, interface circuits, standard I/O Interfaces.

TEXT BOOKS:

REFERENCES:
Unit-I: p-n Junction Diode
Qualitative Theory of p-n Junction, p-n Junction as a Diode, Diode Equation, Volt-Ampere Characteristics, Temperature dependence of VI characteristic, Ideal versus Practical – Resistance levels (Static and Dynamic), Transition and Diffusion Capacitances, Diode Equivalent Circuits, Load Line Analysis, Breakdown Mechanisms in Semi Conductor Diodes, Zener Diode Characteristics.

Unit-II: Rectifiers and Filters
The p-n junction as a Rectifier, Half wave Rectifier, Full wave Rectifier, Bridge Rectifier, Harmonic components in a Rectifier Circuit, Inductor Filters, Capacitor Filters, L- Section Filters, π - Section Filters, Comparision of Filters, Voltage Regulation using Zener Diode.

Unit-III: Bipolar Junction Transistor

Unit-IV: Transistor Biasing and Stabilization
Operating Point, The DC and AC Load lines, Need for Biasing, Fixed Bias, Collector Feedback Bias, Emitter Feedback Bias, Collector - Emitter Feedback Bias, Voltage Divider Bias, Bias Stability, Stabilization Factors, Stabilization against variations in \( V_{BE} \) and \( \beta \), Bias Compensation using Diodes and Transistors, Thermal Runaway, Thermal Stability.

Unit-V: Small Signal Low Frequency BJT Models
BJT Hybrid Model, Determination of h-parameters from Transistor Characteristics, Analysis of a Transistor Amplifier Circuit using h-Parameters, Comparison of CB, CE, and CC Amplifier Configurations.

Unit-VI: Field Effect Transistor

References
3. Introduction to Electronic Devices and Circuits - Rober T. Paynter, PE.
UNIT I
Introduction to Electrical Engineering: ohm’s law, basic circuit components, Kirchhoff’s laws. Simple problems.

UNIT II
Network Analysis: Basic definitions, types of elements, types of sources, resistive networks, inductive networks, capacitive networks, series parallel circuits, star delta and delta star transformation. Network theorems- Superposition, Thevenin’s, Maximum power transfer theorems and simple problems.

UNIT III
Alternating Quantities: Principle of ac voltages, waveforms and basic definitions, root mean square and average values of alternating currents and voltage, form factor and peak factor, phasor representation of alternating quantities, the J operator and phasor algebra, analysis of ac circuits with single basic network element, single phase series circuits.

UNIT IV
Transformers: Principles of operation, Constructional Details, Ideal Transformer and Practical Transformer, Losses, Transformer Test, Efficiency and Regulation Calculations (All the above topics are only elementary treatment and simple problems).

UNIT VI
D.C Generators: Principle of operation of dc machines, types of D.C generators, e.m.f equation in D.C generator.

UNIT VII
D.C motors: Principle of operation of dc motors, types of D.C motors, losses and torque equation, losses and efficiency calculation in D.C generator

UNIT VIII
A.C Machines: Three phase induction motor, principle of operation, slip and rotor frequency, torque (simple problems).

TEXT BOOKS:
1. Basic Electrical Engineering - By M.S.Naidu and S. Kamakshiah – TMH.
3. Electrical and Electronic Technology-By Hughes – Pearson Education.

REFERENCES:
ELECTRICAL AND ELECTRONICS LAB

PART - A
1. Verification of Superposition and Reciprocity theorems.
2. Verification of maximum power transfer theorem. Verification on DC with Resistive load.
3. Experimental determination of Thevenin’s theorem.
5. Swinburne’s Test on DC shunt machine (Predetermination of efficiency of a given DC Shunt machine working as motor and generator).
7. OC & SC tests on Single-phase transformer (Predetermination of efficiency and regulation at given power factors).
8. Brake test on 3-phase Induction motor (performance characteristics).

PART - B
1. PN Junction Diode Characteristics (Forward bias, Reverse bias)
2. Zener Diode Characteristics
3. Transistor CE Characteristics (Input and Output)
4. Rectifier without Filters (Full wave & Half wave)
5. Rectifier with Filters (Full wave & half wave)
Objectives:
- To make the student learn an object oriented way of solving problems.
- To make the student write ADTS for all data structures.

Recommended Systems/Software Requirements:
- Intel based desktop PC with minimum of 166 MHZ or faster processor with atleast 64 MB RAM and 100 MB free disk space
- C++ compiler and STL Recommended

Week 1:
Write C++ programs to implement the following using an array.
  a) Stack ADT
  b) Queue ADT

Week 2:
Write C++ programs to implement the following using a singly linked list.
  a) Stack ADT
  b) Queue ADT

Week 3:
Write C++ programs to implement the deque (double ended queue) ADT using a doubly linked list and an array.

Week 4:
Write a C++ program to perform the following operations:
  a) Insert an element into a binary search tree.
  b) Delete an element from a binary search tree.
  c) Search for a key element in a binary search tree.

Week 5:
Write C++ programs that use recursive functions to traverse the given binary tree in
  a) Preorder
  b) Inorder
  c) Postorder.

Week 6:
Write C++ programs that use non-recursive functions to traverse the given binary tree in
  a) Preorder
  b) Inorder
  c) Postorder.

Week 7:
Write C++ programs for the implementation of bfs and dfs for a given graph.

Week 8:
Write C++ programs for implementing the following sorting methods:
  a) Merge sort
  b) Heap sort

Week 9:
Write a C++ program to perform the following operations
  a) Insertion into a B-tree
  b) Deletion from a B-tree

Week 10:
Write a C++ program to perform the following operation
  a) Insertion into an AVL-tree

Week 11:
Write a C++ program to implement all the functions of a dictionary (ADT) using hashing.

Week 12:
Write a C++ program for implementing Knuth-Morris-Pratt pattern matching algorithm.

(Note: Use Class Templates in the above Programs)

TEXT BOOKS:
5. The Art, Philosophy, and Science of OOP with C++, Rick Miller, SPD.
J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY  
(AUTONOMOUS)  

II Year B.Tech. IT -II Sem          L         T/P/D    C  
4         1/-/-            4  

PRINCIPLES OF PROGRAMMING LANGUAGES  

UNIT I  

UNIT II  
Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.  

UNIT III  
Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization.  

UNIT IV  
Expressions and Statements: Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.  

UNIT-V  
Subprograms and Blocks: Fundamentals of sub-programs, Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are sub-program names, design issues for functions user defined overloaded operators, co routines.  

UNIT VI  
Abstract Data types: Abstractions and encapsulation, introductions to data abstraction, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95 Concurrency: Subprogram level concurrency, semaphores, monitors, massage passing, Java threads, C# threads.  

UNIT VII  
Exception handling : Exceptions, exception Propagation, Exception handler in Ada, C++ and Java.  
Logic Programming Language : Introduction and overview of logic programming, basic elements of prolog, application of logic programming.  

UNIT VIII  
Functional Programming Languages: Introduction, fundamentals of FPL, LISP, ML, Haskell, application of Functional Programming Languages and comparison of functional and imperative Languages.  

TEXT BOOKS:  

REFERENCE BOOKS:  
3. LISP, Patric Henry Winston and Paul Horn, Pearson Education.  
UNIT I:

UNIT II:
History of Data base Systems. Data base design and ER diagrams – Beyond ER Design Entities, Attributes and Entity sets – Relationships and Relationship sets – Additional features of ER Model – Concept Design with the ER Model – Conceptual Design for Large enterprises.

UNIT III:

UNIT IV:
Form of Basic SQL Query – Examples of Basic SQL Queries – Introduction to Nested Queries – Correlated Nested Queries Set – Comparison Operators – Aggregative Operators – NULL values – Comparison using Null values – Logical connectivity’s – AND, OR and NOT – Impact on SQL Constructs – Outer Joins – Disallowing NULL values – Complex Integrity Constraints in SQL Triggers and Active Data bases.

UNIT V:

UNIT VI:

UNIT VII:

UNIT VIII:

TEXT BOOKS:

REFERENCES:
2. Fundamentals of Database Systems, Elmasri Navrate Pearson Education
3. Introduction to Database Systems, C.J.Date Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V.Shah,SPD.
5. Database Systems Using Oracle:A Simplified guide to SQL and PL/SQL, Shah,PHI.
UNIT I:
Object oriented thinking: Need for oop paradigm, A way of viewing world – Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, overriding and exceptions, summary of oop concepts, coping with complexity, abstraction mechanisms.

UNIT II:
Java Basics: History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT III:
Inheritance – Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class.

UNIT IV:
Packages and Interfaces: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT V:
Exception handling: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling, Exploring java.util.

UNIT VI:
Multithreading: Differences between multi threading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, interthread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT VII:
Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components – labels, button, canvas, scrollbars, text components, check box, check box groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

UNIT VIII:

TEXT BOOKS:
1. Java: the complete reference, 7th edition, Herbert schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, pearson eduction.

REFERENCES:
2. An Introduction to OOP, third edition, T. Budd, pearson education.
3. Introduction to Java programming, Y. Daniel Liang, pearson education.
9. Maurach’s Beginning Java2 JDK 5 , SPD.
UNIT-I: ECOSYSTEMS: Concept of ecosystem, Classification of ecosystem, Functions of ecosystem, Food chains, Food webs and ecological pyramids, Flow of energy, Biogeochemical cycles, Biomagnification, carrying capacity.

UNIT-II: NATURAL RESOURCES: Classification of Resources: Living and Non-Living resources, Renewable and Non-Renewable resources. Water resources: use and over utilization, Land resources, land degradation, Forest resources, Mineral resources uses. Energy resources: growing energy needs, use of alternate energy sources-case studies. Environmental effects due to exploitation of various resources.

UNIT-III: BIODIVERSITY AND BIOTIC RESOURCES: Species, ecosystem diversity, Hotspots, Value of biodiversity, Threats to biodiversity, Conservation of biodiversity: In-Situ and Ex-Situ conservation, Biological disasters, pandemic and epidemics, Biological warfare.

UNIT-IV: ENVIRONMENTAL POLLUTION AND CONTROL: Classification of pollutions and pollutants, causes, effects of water, air, noise pollution, Introduction to control technologies: Water (primary, secondary, tertiary), Air(particulate and gaseous emissions), Soil(conservation and remediation), Noise(controlling devices) Solid waste : types, collection and disposal methods, characteristics of e-waste and its management.


Text Book:
1. TEXT BOOK OF ENVIRONMENTAL Science and Technology by M.Anji Reddy 2007
2. Principles of Environmental Science and Engineering by P.Venugopal Rao
3. Introduction to Environmental Studies by K.Mukkanti

References
1. Tata McgrawHill : Introduction to Environmental Studies by Benny Joseph
2. Environmental studies by Erach Bharucha 2005, University Grants Commission, University Press
UNIT-I : Probability:

UNIT-II: Distributions
Binomial, Poisson & normal distributions related properties. Sampling distributions – Sampling distribution of means ($\sigma$ known and Unknown)

UNIT-III: Testing of Hypothesis I
Tests of hypothesis point estimations – interval estimations Bayesian estimation. Large samples, Null hypothesis – Alternate hypothesis type I, & type II errors – critical region confidential interval for mean testing of single variance. Difference between the mean.

UNIT-IV : Testing of Hypothesis II
Confidential interval for the proportions. Tests of hypothesis for the proportions single and difference between the proportions.

UNIT-V: Small samples

UNIT-VI: Correlation & Regression
Coefficient of correlation – Regression Coefficient – The lines of regression – The rank correlation

UNIT-VII: Queuing Theory
Arrival Theorem - Pure Birth process and Death Process M/M/1 Model. MATLAB/R Introduction.

UNIT-VIII: Stochastic processes

TEXT BOOKS:
3. Introduction to MATLAB by RudraGupta
UNIT I:
Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis - Space complexity, Time complexity, Asymptotic Notation - Big oh notation, Omega notation, Theta notation and Little oh notation, Probabilistic analysis, Amortized analysis.

UNIT II:
Disjoint Sets - disjoint set operations, union and find algorithms, spanning trees, connected components and biconnected components.

UNIT III:
Divide and conquer: General method, applications - Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT IV:
Greedy method: General method, applications - Job sequencing with dead lines, 0/1 knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT V:
Dynamic Programming: General method, applications - Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

UNIT VI:
Backtracking: General method, applications - n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT VII:
Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem, LC Branch and Bound solution, FIFO Branch and Bound solution.

UNIT VIII:
NP-Hard and NP-Complete problems: Basic concepts, non deterministic algorithms, NP - Hard and NPComplete classes, Cook's theorem.

TEXT BOOKS:
2. Design and Analysis Algorithms - Parag Himanshu Dave, Himanshu Bhalchandra Dave Publisher: Pearson

REFERENCES:
5. Algorithms – Richard Johnsonbaugh and Marcus Schaefer, Pearson Education
OBJECT ORIENTED PROGRAMMING LAB

Objectives:
- To make the student learn an object oriented way of solving problems.
- To teach the student to write programs in Java to solve the problems

Recommended Systems/Software Requirements:
- Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space
- JDK Kit. Recommended

Week 1:
- a) Write a Java program that prints all real solutions to the quadratic equation \( ax^2 + bx + c = 0 \). Read in \( a \), \( b \), \( c \) and use the quadratic formula. If the discriminant \( b^2 - 4ac \) is negative, display a message stating that there are no real solutions.
- b) The Fibonacci sequence is defined by the following rule:
   The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non-recursive functions to print the \( n \)th value in the Fibonacci sequence.

Week 2:
- a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
- b) Write a Java program to multiply two given matrices.
- c) Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use StringTokenizer class of java.util)

Week 3:
- a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
- b) Write a Java program for sorting a given list of names in ascending order.
- c) Write a Java program to make frequency count of words in a given text.

Week 4:
- a) Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
- b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
- c) Write a Java program that displays the number of characters, lines and words in a text file.

Week 5:
- a) Write a Java program that:
  - i) Implements a stack ADT.
  - ii) Converts an infix expression into Postfix form.
  - iii) Evaluates the postfix expression.

Week 6:
- a) Develop an applet that displays a simple message.
- b) Develop an applet that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named “Compute” is clicked.

Week 7:
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.

Week 8:
- a) Write a Java program for handling mouse and key events.

Week 9:
- a) Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.
- b) Write a Java program that correctly implements the producer consumer problem using the concept of inter-thread communication.
Week 10:
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 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 NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Week 11:
a) Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.
b) Write a Java program that allows the user to draw lines, rectangles, and ovals.

Week 12:
a) Write a java program to create an abstract class named Shape that contains an empty method named numberOfSides(). Provide three classes named Trapezoid, Triangle, and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides() that shows the number of sides in the given geometrical figures.
b) 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. Write a java program to display the table using JTable component.

TEXT BOOKS:

2. Programming with Java, M.P.Bhave and S.A.Patekar, Pearson Education
7. Essentials of Java Programming, Muthu,C, TMH.
9. The Art, Philosophy, and Science of OOP with Java, R.Miller, R.Kasprian, SPD.
Objective: This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example company named “Roadway Travels” whose description is as follows. The student is expected to practice the designing, developing and querying a database in the context of example database “Roadway travels”. Students are expected to use “Mysql” database.

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 following areas:
- Reservations and Ticketing
- Cancellations

Reservations & 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 query’s using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 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

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the 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.

Note: The student is required to submit a document by writing the Entities and Keys to the lab teacher.

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
**Note:** The student is required to submit a document by drawing the E-R Diagram to the lab teacher.

### 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.

<table>
<thead>
<tr>
<th>Passenger</th>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Ticket_id</th>
<th>Passport ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.

### 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 within 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 along with the primary key of passenger.

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

<table>
<thead>
<tr>
<th>Passenger</th>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Passport ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Passport ID</th>
<th>Ticket_id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You can do the second and third normal forms if required. Anyhow Normalized tables are given at the end.

**Experiment 5: Installation of Mysql and practicing DDL commands**

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

Example for creation of a normalized "Passenger" table.

```sql
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 create all other tables.

**Note: Detailed creation of tables is given at the end.**

**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 table
- **UPDATE** - 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 you 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 and 45.
6. Display all the passengers names beginning with ‘A’
7. Display the sorted list of passengers names

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

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

1. Write a Query to display the Information present in the Passenger and cancellation tables. Hint: Use UNION Operator.
2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. Hint: Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. Hint: Use GROUP BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

Experiment 10: Triggers

In this week you are going to work on 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 ROW
BEGIN
  IF NEW.TicketNO > 60 THEN
    SET New.Ticket no = Ticket no;
  ELSE
    SET New.Ticketno = 0;
  END IF;
END;

Experiment 11: Procedures

In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

Eg: CREATE PROCEDURE myProc()
BEGIN
SELECT COUNT(Tickets) FROM Ticket WHERE age>=40;
End;

Experiment 12: Cursors

In this week you need to do the following: 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_id   INT;
DECLARE v_name VARCHAR(30);
DECLARE c1 CURSOR FOR  SELECT stdId,stdFirstname FROM students WHERE stdId=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 : Varchar

Passenger
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) : 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 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 other than 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

REFERENCE BOOKS:
1. Introduction to SQL, Rick F. Vander Lans, Pearson education.
3. Oracle PL/SQL Programming, Steven Feuerstein, SPD.
4. SQL & PL/SQL for Oracle 10g, Black Book, Dr. P.S. Deshpande, Dream Tech.
5. Oracle Database 11g PL/SQL Programming, M. Mc Laughlin, TMH.
6. SQL Fundamentals, J.J. Patrick, Pearson Education.