## Course Structure and Syllabus

### I Year – I Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core Course I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Structures and Algorithms</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database Internals</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course III</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed Systems</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Elective I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Network Security</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>2. Android Application Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Cloud Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Internet of Things</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Core Elective II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Machine Learning</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>2. Parallel and Distributed Algorithms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Software Architecture and Design Patterns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Embedded Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Open Elective I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Elective – 1</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Laboratory I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Structures and Algorithms Lab</td>
<td>25</td>
<td>75</td>
<td>--</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td><strong>Seminar I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seminar</td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td>24</td>
<td>8</td>
<td>28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### I Year – II Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core Course IV</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Programming</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course V</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Retrieval Systems</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Course VI</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet Technologies and Services</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Core Elective III</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core Elective – 3</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>1. Data Mining</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Storage Area Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Semantic Web and Social Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Cyber Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Core Elective IV</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core Elective – 4</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td>1. Big Data Analytics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Soft Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Software Process and Project Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Distributed Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Open Elective II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Elective – 2</td>
<td>25</td>
<td>75</td>
<td>4</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Laboratory II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet Technologies and Services Lab</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td><strong>Seminar II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seminar</td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td>24</td>
<td>8</td>
<td>28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### II Year - I Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comprehensive Viva-Voce</strong></td>
<td>--</td>
<td>100</td>
<td>--</td>
<td>--</td>
<td>4</td>
</tr>
<tr>
<td><strong>Project work Review I</strong></td>
<td></td>
<td></td>
<td>50</td>
<td>--</td>
<td>12</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
</tr>
</tbody>
</table>

### II Year - II Semester

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Int. marks</th>
<th>Ext. marks</th>
<th>L</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project work Review II</strong></td>
<td>50</td>
<td>--</td>
<td>--</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td><strong>Project Evaluation (Viva-Voce)</strong></td>
<td>--</td>
<td>150</td>
<td>--</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
</tr>
</tbody>
</table>
Open Electives

1. Basic Computer Programming skills are required for all open electives. Additionally, knowledge on the specified area mentioned in prerequisites is required for opting the open elective.

2. Note: A student can register for any open elective subject provided that he has not already registered for the same subject.

<table>
<thead>
<tr>
<th>S.NO</th>
<th>Open Electives</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>&quot;R&quot; Programming</td>
<td>Maths, Statistics</td>
</tr>
<tr>
<td>2.</td>
<td>Android Application Development</td>
<td>Java</td>
</tr>
<tr>
<td>3.</td>
<td>Algorithmics</td>
<td>----</td>
</tr>
<tr>
<td>4.</td>
<td>Big Data Analytics</td>
<td>Data Bases, Maths</td>
</tr>
<tr>
<td>5.</td>
<td>Bioinformatics</td>
<td>Data Structures</td>
</tr>
<tr>
<td>6.</td>
<td>Biometrics</td>
<td>----</td>
</tr>
<tr>
<td>7.</td>
<td>Cyber Security</td>
<td>Internet Technologies</td>
</tr>
<tr>
<td>8.</td>
<td>Computer Forensics</td>
<td>Maths, Data Structures</td>
</tr>
<tr>
<td>10.</td>
<td>E-Commerce</td>
<td>Internet Technologies</td>
</tr>
<tr>
<td>11.</td>
<td>Embedded Systems</td>
<td>Digital logic</td>
</tr>
<tr>
<td>12.</td>
<td>Information Security</td>
<td>Maths</td>
</tr>
<tr>
<td>13.</td>
<td>Intellectual Property Rights</td>
<td>----</td>
</tr>
<tr>
<td>14.</td>
<td>Internet of Things</td>
<td>Java</td>
</tr>
<tr>
<td>15.</td>
<td>Java Programming</td>
<td>----</td>
</tr>
<tr>
<td>16.</td>
<td>Linux Programming</td>
<td>----</td>
</tr>
<tr>
<td>17.</td>
<td>Mobile Computing</td>
<td>Java</td>
</tr>
<tr>
<td>18.</td>
<td>Mobile Application Security</td>
<td>Mobile Application Development</td>
</tr>
<tr>
<td>19.</td>
<td>OpenStack cloud computing</td>
<td>Linux Programming</td>
</tr>
<tr>
<td>20.</td>
<td>Operations Research</td>
<td>Maths, Data Structures</td>
</tr>
<tr>
<td>22.</td>
<td>Scripting Languages</td>
<td>----</td>
</tr>
<tr>
<td>23.</td>
<td>Social Media Intelligence</td>
<td>----</td>
</tr>
<tr>
<td>24.</td>
<td>Software Engineering</td>
<td>----</td>
</tr>
<tr>
<td>25.</td>
<td>Storage Area Networks</td>
<td>Computer Networks</td>
</tr>
<tr>
<td>26.</td>
<td>Web Usability</td>
<td>----</td>
</tr>
</tbody>
</table>
DATA STRUCTURES AND ALGORITHMS

Objectives:
The fundamental design, analysis, and implementation of basic data structures. Basic concepts in the specification and analysis of programs. Principles for good program design, especially the uses of data abstraction. Significance of algorithms in the computer field. Various aspects of algorithm development. Qualities of a good solution.

UNIT I
Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation—Big Oh, Omega and Theta notations, Complexity Analysis Examples.
Data structures—Linear and non-linear data structures, ADT concept, Linear List ADT, Array representation, Linked representation, Vector representation, singly linked lists—insertion, deletion, search operations, doubly linked lists—insertion, deletion operations, circular lists. Representation of single, two dimensional arrays, Sparse matrices and their representation.

UNIT II
Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack, implementation of recursion, Circular queue-insertion and deletion, Desueteude ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap, java.util package-ArrayList, LinkedList, Vector classes, Stacks and Queues in java.util, Iterators in java.util.

UNIT III

UNIT IV
Trees—Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree ADT, representations, recursive and non-recursive traversals, Java code for traversals, Threaded binary trees.
Graphs—Graphs terminology, Graph ADT, representations, graph traversals/search methods—dfs and bfs, Java code for graph traversals, Applications of Graphs—Minimum cost spanning tree using Kruskal’s algorithm, Dijkstra’s algorithm for Single Source Shortest Path Problem.

UNIT V
Search trees—Binary search tree—Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees—Definition and examples only, Red Black trees—Definition and examples only, B-Trees—Definition, insertion and searching operations, Trees in java.util—TreeSet, Tree Map Classes, Trie(examples only), Comparison of Search trees.
Text compression—Huffman coding and decoding, Pattern matching—KMP algorithm.

TEXT BOOKS:
1. Data structures, Algorithms and Applications in Java, S.Sahni, Universities Press.

REFERENCE BOOKS:
1. Java for Programmers, Deitel and Deitel, Pearson education.
6. Classic Data structures in Java, T.Budd, Addison-Wesley (Pearson Education).
7. Data structures with Java, Ford and Topp, Pearson Education.
OBJECTIVES:

By the end of the course, you will know:

- History and Structure of databases
- How to design a database
- How to convert the design into the appropriate tables
- Handling Keys appropriately
- Enforcing Integrity Constraints to keep the database consistent
- Normalizing the tables to eliminate redundancies
- Querying relational data and processing the queries
- Triggers, Procedures and Cursors, Transaction Management
- Distributed databases management system concepts and Implementation

UNIT I

Database System Applications, Purpose of Database Systems, View of Data – Data Abstraction, Instances and Schemas, Data Models – the ER Model, Relational Model, Other Models – Database Languages – DDL, DML, Database Access from Applications Programs, Transaction Management, Data Storage and Querying, Database Architecture, Database Users and Administrators, ER diagrams,. Relational Model: Introduction to the Relational Model – Integrity Constraints Over Relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design, Introduction to Views – Altering Tables and Views, Relational Algebra, Basic SQL Queries, Nested Queries, Complex Integrity Constraints in SQL, Triggers

UNIT II


UNIT III

Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions – Lock Based Concurrency Control, Deadlocks – Performance of Locking – Transaction Support in SQL.

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

Crash recovery: Introduction to Crash recovery, Introduction to ARIES, the Log, and Other Recovery related Structures, the Write-Ahead Log Protocol, Check pointing, recovering from a System Crash, Media recovery

UNIT IV

Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing – Clustered Indexes, Primary and Secondary Indexes, Index data Structures – Hash Based Indexing, Tree based Indexing


Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendable Vs Linear Hashing.

UNIT V

Distributed databases: Introduction to distributed databases, Distributed DBMS architectures, Storing data in a distributed DBMS, Distributed catalog management, Distributed query processing Updating distributed data, Distributed transactions, Distributed concurrency control, Distributed recovery

TEXT BOOKS:

REFERENCE BOOKS:

1. Introduction to Database Systems, C.J.Date, Pearson Education.
2. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, PHI.
9. Distributed Databases Principles & Systems, Stefano Ceri, Giuseppe Pelagatti, TMH.
J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY

M. Tech-CSE – I Year – I Sem

DISTRIBUTED SYSTEMS

Objectives:

- Understand the need for distributed systems and their applications.
- Understand the concepts of remote procedure calls, remote file systems, distributed agreement, clock synchronization, and security.

UNIT I


Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

UNIT II


Name Services-Introduction, Name Services and the Domain Name System, Case study of the Global Name Service, Case study of the X.500 Directory Service.

UNIT III

Peer to Peer Systems–Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays, Overlay case studies-Pastry, Tapestry, Application case studies-Squirrel, Ocean Store, Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement-Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT IV

Transactions and Concurrency control-Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control. Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication-Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data.

UNIT V

Security-Introduction, Overview of Security techniques, Cryptographic algorithms, Digital signatures, Case studies-Kerberos, TLS, 802.11 Wi-Fi.

Distributed shared memory, Design and Implementation issues, Sequential consistency and Ivy case study, Release consistency and Munin case study, Other consistency models, CORBA case study-Introduction, CORBA RMI, CORBA Services.

TEXT BOOKS:


REFERENCE BOOKS:

Objectives:

- Understand the basic categories of threats to computers and networks
- Understand various cryptographic algorithms.
- Describe public-key cryptosystem.
- Describe the enhancements made to IPv4 by IPSec
- Understand Intrusions and intrusion detection
- Discuss the fundamental ideas of public-key cryptography.
- Generate and distribute a PGP key pair and use the PGP package to send an encrypted e-mail message.
- Discuss Web security and Firewalls

UNIT – I


UNIT – II

Symmetric key Ciphers: Block Cipher principles & Algorithms (DES, AES, Blowfish), Differential and Linear Cryptanalysis, Block cipher modes of operation, Stream ciphers, RC4, Location and placement of encryption function, Key distribution Asymmetric key Ciphers: Principles of public key cryptosystems, Algorithms (RSA, Diffie-Hellman, ECC), Key Distribution

UNIT – III


UNIT – IV


UNIT – V


TEXT BOOKS:


REFERENCE BOOKS:

3. Information Security, Principles and Practice : Mark Stamp, Wiley India.
SOFTWARE ARCHITECTURE AND DESIGN PATTERNS
(CORE ELECTIVE –II)

Objectives:
After completing this course, the student should be able to:

- To understand the concept of patterns and the Catalog.
- To discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency.
- To understand the variety of implemented bad practices related to the Business and Integration tiers.
- To highlight the evolution of patterns.
- To how to add functionality to designs while minimizing complexity
- To understand what design patterns really are, and are not
- To learn about specific design patterns.
- To learn how to use design patterns to keep code quality high without overdesign.

UNIT I
Envisioning Architecture

Creating an Architecture
Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

UNIT II
Analyzing Architectures
Architecture Evaluation, Architecture design decision making, ATAM, CBAM.

Moving from one system to many
Software Product Lines, Building systems from off the shelf components, Software architecture in future.

UNIT III
Patterns
Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage.

Creational and Structural patterns
Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight.

UNIT IV
Behavioral patterns
Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

UNIT V
Case Studies

TEXT BOOKS:

REFERENCE BOOKS:
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
5. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
Objectives:

- To familiar with constructors and string handling functions
- To explain Inheritance and Polymorphism
- To familiar with Packages and Interfaces
- To familiar with Exception handling and Multithreading
- To familiar with Applet Programming, Event Handling and scripting.

Outcomes:

- Be familiar with constructors and string handling
- Able to understand Inheritance and Polymorphism
- Be able to understand Packages and Interfaces
- Be able to understand Exception handling and Multithreading
- Be able to understand Applet Programming
- Be able to implement Event Handling

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.

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 II:

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.

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 III:

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

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 IV :

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 V :

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.


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.
J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY

M. Tech-CSE – I Year – I Sem
DATA STRUCTURES AND ALGORITHMS LAB

Objectives:

- The fundamental design, analysis, and implementation of basic data structures.
- Basic concepts in the specification and analysis of programs.
- Principles for good program design, especially the uses of data abstraction.

Sample Problems on Data structures:

1. Write Java programs that use both recursive and non-recursive functions for implementing the following searching methods:
   a) Linear search  
   b) Binary search

2. Write Java programs to implement the following using arrays and linked lists
   a) List ADT

3. Write Java programs to implement the following using an array.
   a) Stack ADT  
   b) Queue ADT

4. Write a Java program that reads an infix expression and converts the expression to postfix form. (Use stack ADT).

5. Write a Java program to implement circular queue ADT using an array.

6. Write a Java program that uses both a stack and a queue to test whether the given string is a palindrome or not.

7. Write Java programs to implement the following using a singly linked list.
   a) Stack ADT  
   b) Queue ADT

8. Write Java programs to implement the deque (double ended queue) ADT using a) Array b) Singly linked list c) Doubly linked list.

9. Write a Java program to implement priority queue ADT.

10. Write a Java program to perform the following operations:
    a) Construct a binary search tree of elements.
    b) Search for a key element in the above binary search tree.
    d) Delete an element from the above binary search tree.

11. Write a Java program to implement all the functions of a dictionary (ADT) using Hashing.

12. Write a Java program to implement Dijkstra’s algorithm for Single source shortest path problem.

13. Write Java programs that use recursive and non-recursive functions to traverse the given binary tree in
    a) Preorder  
    b) Inorder  
    c) Postorder.

14. Write Java programs for the implementation of bfs and dfs for a given graph.

15. Write Java programs for implementing the following sorting methods:
    a) Bubble sort  
    d) Merge sort  
    g) Binary tree sort
    b) Insertion sort  
    e) Heap sort
    c) Quick sort  
    f) Radix sort

16. Write a Java program to perform the following operations:
    a) Insertion into a B-tree  
    b) Searching in a B-tree

17. Write a Java program that implements Kruskal’s algorithm to generate minimum cost spanning tree.

18. Write a Java program that implements KMP algorithm for pattern matching.
REFERENCE BOOKS:

2. Data Structures with Java, J.R.Hubbard, 2nd edition, Schaum's Outlines, TMH.

(Note: Use packages like java.io, java.util, etc)