Savitribai Phule Pune University

(Formerly University of Pune)

Three Year B.Sc. Degree Program in Computer Science

(Faculty of Science & Technology)

S.Y.B.Sc. (Computer Science)

Choice Based Credit System Syllabus
To be implemented from Academic Year
2020-2021

S. Y. B. Sc.(Computer Science)

Semester III

(Total credits=22)

Course type	Paper Code	Paper title	Credits	Evaluation		
				CA	UE	TOTAL
CC-VIII	CS 231	Data Structures and Algorithms – I	2	15	35	50
	CS 232	Software Engineering	2	15	35	50
	CS 233	Practical course on CS 231 and CS 232	2	15	35	50
CC-IX		Mathematics - I	2	15	35	50
		Mathematics - II	2	15	35	50
		Practical course in Mathematics	2	15	35	50
CC-X		Electronics - I	2	15	35	50
		Electronics - II	2	15	35	50
		Practical course in Electronics	2	15	35	50
AECC-I		Environment Science – I	2			
AECC-II		Language Communication – I	2			

Semester IV

(Total credits=22)

Course type	Paper Code	Paper title	Credits	Evaluation		
				CA	UE	TOTAL
CC-XI	CS 241	Data Structures and Algorithms – II	2	15	35	50
	CS 242	Computer Networks - I	2	15	35	50
	CS 243	Practical course on CS 241 and CS 242	2	15	35	50
CC-XII		Mathematics - I	2	15	35	50
		Mathematics - II	2	15	35	50
		Practical course in Mathematics	2	15	35	50
CC-XIII		Electronics - I	2	15	35	50
		Electronics - II	2	15	35	50
		Practical course in Electronics	2	15	35	50
AECC-I		Environment Science – II	2			
AECC-II		Language Communication –II	2			

- Each theory Lecture time for S.Y. B.Sc Computer Science is of 50 min (3 lectures/ week for 2 credit course)
- Each practical session time for S.Y. B.Sc Computer Science is of 4 hrs 20 minutes (260 min)
- Practical batch size =12

Savitribai Phule Pune University
S.Y.B.Sc. (Computer Science)
Semester – I
Computer Science Paper – I
Course Code: CS 231
Title : Data Structures and Algorithms – I

Teaching Scheme	No. of Credits	Examination Scheme
3 Lectures / week (50 mins duration)	2	IE : 15 marks UE: 35 marks

Prerequisites:
Basic knowledge of algorithms and problem solving
Knowledge of C Programming Language

Course Objectives
1. To learn the systematic way of solving problem
2. To understand the different methods of organizing large amount of data
3. To efficiently implement the different data structures
4. To efficiently implement solutions for specific problems
5. To apply linear data structures.

Course Outcomes: On completion of the course, student will be able to
1. To use well-organized data structures in solving various problems.
2. To differentiate the usage of various structures in problem solution.
3. Implementing algorithms to solve problems using appropriate data structures.

Course Contents

Chapter 1	**Introduction to Data Structures and Algorithm Analysis**	**4 lectures**

1.1 Introduction
- 1.1.1 Need of Data Structure
- 1.1.2 Definitions - Data and information, Data type, Data object, ADT, Data Structure
- 1.1.3 Types of Data Structures

1.2 Algorithm analysis
1.2.1 Space and time complexity, Graphical understanding of the relation between different functions of n, examples of linear loop, logarithmic,quadratic loop etc.
1.2.2 Best, Worst, Average case analysis, Asymptotic notations (Big O, Omega Ω, Theta θ), Problems on time complexity calculation.

Chapter 2	**Array as a Data Structure**	**10 lectures**

2.1 ADT of array, Operations
2.2Array applications - Searching
- 2.2.1 Sequential search, variations - Sentinel search, Probability search, ordered list search
- 2.2.2 Binary Search
- 2.2.3 Comparison of searching methods

2.3 Sorting Terminology- Internal, External, Stable, In-place Sorting

2.3.1 Comparison Based Sorting - Lower bound on comparison based sorting, Methods- Bubble Sort, Insertion Sort, Selection Sort, Algorithm design strategies - Divide and Conquer strategy, Merge Sort, Quick Sort, complexity analysis.

2.3.2 Non Comparison Based Sorting: Counting Sort, Radix Sort, complexity analysis.

2.3.3 Comparison of sorting methods

Chapter 3	**Linked List**	**10 lectures**

3.1 List as a Data Structure, differences with array.
3.2 Dynamic implementation of Linked List, internal and external pointers
3.3 Types of Linked List – Singly, Doubly, Circular
3.4 Operations on Linked List - create, traverse, insert, delete, search, sort, reverse, concatenate, merge, time complexity of operations.
3.5 Applications of Linked List – polynomial representation, Addition of two polynomials
3.6 Generalized linked list – concept, representation, multiple-variable polynomial representation using generalized list.

Chapter 4	**Stack**	**6 lectures**

4.1 Introduction
4.2 Operations – init(), push(), pop(), isEmpty(), isFull(), peek(), time complexity of operations.
4.3 Implementation- Static and Dynamic with comparison
4.4 Applications of stack

4.4.1 Function call and recursion, String reversal, palindrome checking

4.4.2 Expression types - infix, prefix and postfix, expression conversion and evaluation (implementation of infix to postfix, evaluation of postfix)

4.4.3Backtracking strategy - 4 queens problem (implementation using stack)

Chapter 5	**Queue**	**6 lectures**

5.1 Introduction
5.2 Operations - init(), enqueue(), dequeue(), isEmpty(), isFull(), peek(),time complexity of operations, differences with stack.
5.3 Implementation - Static and Dynamic with comparison
5.4 Types of Queue - Linear Queue, Circular Queue, Priority Queue, Double Ended Queue (with implementation)
5.5 Applications – CPU Scheduling in multiprogramming environment, Round robin algorithm

Reference Books:

1. Fundamentals of Data Structures in C- Ellis Horowitz, Sartaj Sahni,Susan Anderson-Freed, 2nd Edition, Universities Press.
2. Data Structures using C and C++-Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
3. Classic Data Structures-D. Samanta, Prentice Hall India Pvt. Ltd.
4. Data Structures: A Pseudo code approach with C, Richard Gilberg, Behrouz A. Forouzan, Cengage Learning.
5. Introduction to Algorithms—Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein--MIT Press
6. Data Structures and Program Design in C- 2nd Edition, Robert L Kruse, Bruce P. Leung, Clovis L. Tondo, Pearson Education.

Savitribai Phule Pune University
S.Y.B.Sc. (Computer Science)
Semester – I
Computer Science Paper -II
Course Code: CS 232
Title : Software Engineering

Teaching Scheme	No. of Credits	Examination Scheme
3 lectures / week (50 mins duration)	**2**	IE : 15 marks UE: 35 marks

Prerequisites
ER Modeling

Course Objectives

1. To get knowledge and understanding of software engineering discipline.
2. To learn analysis and design principles for software project development.

Course Outcomes
On completion of the course, student will be able to-

1. Compare and chose a process model for a software project development.
2. Identify requirements analyze and prepare models.
3. Prepare the SRS, Design document, Project plan of a given software system.

Course Contents

Chapter 1	**Title : Introduction To Software Engineering and Process Models**	**8 lectures**

1.1 Definition of Software
1.2 Nature of Software Engineering
1.3 Changing nature of software
1.4 Software Process
- 1.4.1 The Process Framework
- 1.4.2 Umbrella Activities
- 1.4.3 Process Adaptation

1.5 Generic Process Model
1.6 Prescriptive Process Models
- 1.6.1 The Waterfall Model
- 1.6.2 Incremental Process Models
- 1.6.3 Evolutionary Process Models
- 1.6.4 Concurrent Models
- 1.6.5 The Unified Process

Chapter 2	**Title : Agile Development**	**5lectures**

2.1 What is Agility?
2.2 Agile Process
- 2.2.1 Agility Principles
- 2.2.2 The Politics Of Agile Development
- 2.2.3 Human Factors

2.3 Extreme Programming(XP)
- 2.3.1XP Values
- 2.3.2XP Process

2.3.3 Industrial XP
2.4 Adaptive Software Development(ASD)
2.5 Scrum
2.6 Dynamic System Development Model (DSDM)
2.7 Agile Unified Process (AUP)

Chapter 3	**Title : Requirements Analysis**	**7 lectures**

3.1 Requirement Elicitation,
3.2 Software requirement specification (SRS)
3.2.1 Developing Use Cases (UML)
3.3 Building the Analysis Model
3.3.1 Elements of the Analysis Model
3.3.2 Analysis Patterns
3.3.3 Agile Requirements Engineering
3.4 Negotiating Requirements
3.5 Validating Requirements

Chapter 4	**Title : Requirements Modeling**	**10 lectures**

4.1 Introduction to UML
4.1.1 Goals of UML
4.1.2 UML Concept Areas
4.1.3 UML Views
4.1.4 Model and its significance
4.2Structural Modeling
4.2.1 Use case model
4.2.2Class model
4.3Behavioral Modeling
4.3.1 Sequence model
4.3.2 State Machine Model
4.3.3 Activity model
4.3.4 Communication or Collaboration model
4.4 Architectural Modeling
4.4.1 Component model
4.4.2 Deployment model
4.5 Model Management view
4.5.1 Package View, Package diagram
4.6 Extensibility Constructs

Chapter 5	**Title :Design Concepts**	**6lectures**

5.1 Design Process
5.1.1 Software Quality Guidelines and Attributes
5.1.2 Evolution of Software Design
5.2 Design Concepts
5.2.1 Abstraction
5.2.2 Architecture
5.2.3 Patterns
5.2.4 Separation of Concerns
5.2.5 Modularity
5.2.6 Information Hiding
5.2.7 Functional Independence
5.2.8 Refinement

5.2.9 Aspects
5.2.10 Refactoring
5.2.11 Object Oriented Design Concepts
5.2.12 Design Classes
5.2.13 Dependency Inversion
5.2.14 Design for Test
5.3 The Design Model
5.3.1 Data Design Elements
5.3.2 Architectural Design Elements
5.3.3 Interface Design Elements
5.3.4 Component-Level Diagram
5.4.5 Deployment-Level Diagram

Reference Books:

1. Software Engineering : A Practitioner's Approach - Roger S. Pressman, McGraw hill(Eighth Edition) ISBN-13: 978-0-07-802212-8, ISBN-10: 0-07-802212-6
2. A Concise Introduction to Software Engineering - Pankaj Jalote, Springer ISBN: 978-1-84800-301-9
3. The Unified Modeling Language Reference Manual - James Rambaugh, Ivar Jacobson, Grady Booch ISBN 0-201-30998-X
4. The Unified Modeling Language User Guide - James Rambaugh, Ivar Jacobson, Grady Booch ISBN: 0-201-57168-4

Savitribai Phule Pune University
S.Y.B.Sc. (Computer Science)
Semester – I
Computer Science Paper - III
Course Code: CS 233
Title : Practical course on CS 231 (Data Structures and Algorithms I) and CS 232 (Software Engineering)

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs 20 mins / week Batch Size : 12	2	IE : 15 marks UE: 35 marks

Operating Environment:
For Data Structures:

- **Operating system:** Linux
- **Editor:** Any linux based editor like vi, gedit etc.
- **Compiler** : cc or gcc

Lab Book:
The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Programming Assignments:
Programs should be done individually by the student intheir respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

Assessment:
Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include-timely completion, performance, innovation, efficient codes and good programming practices.

- **Internal Evaluation :**
  - 10 marks will be given based on a mini project of Software Engineering.
  - 5 marks will be allocated for Assignment completion and practical attendance.
- **University Evaluation :**
  - The Practical slip will be of 35 Marks which will be based on Data structures.

Course Contents:

Suggested Assignments for Data Structures – I

Assignment 1: Searching Algorithms
Implementation of searching algorithms to search an element using: Linear Search, Variations of linear search, Binary Search (with time complexity)

Assignment 2: Sorting Algorithms - I
Implementation of sorting algorithms: Bubble Sort, Insertion Sort, Selection Sort

Assignment 3: Sorting Algorithms - II

Implementation of sorting algorithms: Quick Sort, Merge Sort

Assignment 4: Non-comparison based Sorting

Implementation of sorting algorithms: Counting Sort, Radix Sort

Assignment 5: Singly Linked List

1. Dynamic implementation of Singly Linked List to perform following operations: Create, Insert, Delete, Display, Search, Reverse
2. Create a list in the sorted order.

Assignment 6: Doubly Linked List

1. Dynamic implementation of Doubly circular Linked List to perform following operations: Create, Insert, Delete, Display, Search

Assignment 7: Linked List Applications

1. Merge two sorted lists.
2. Addition of two polynomials in a single variable.

Assignment 8: Stack

1. Static and Dynamic implementation of Stack to perform following operations: Init, Push, Pop, Peek, Isempty, Isfull.
2. Reverse string of characters using stack and check whether a string is a palindrome.

Assignment 9: Applications of Stack

1. Infix to Postfix conversion and evaluation.
2. 4-Queens algorithm using backtracking

Assignment 10: Linear Queue

1. Static and Dynamic implementation of linear Queue

Assignment 11: Circular and Priority Queue

1. Implementation of circular queue
2. Implementation of priority queue

Suggested Assignments for Software Engineering mini project	**3**

1. Prepare detailed statement of problem for the selected mini project
2. Identify suitable process model for the same.
3. Develop Software Requirement Specification for the mini project.
4. Identify scenarios and develop UML Use case
5. Other artifacts: Class Diagram, activity diagram, sequence diagram, component diagram and any other diagrams as applicable to the mini project.

Sample mini project titles: (These are just samples, students are suggested to take up different case studies)

1. Online mobile recharge system
2. Credit calculation system
3. Image sharing and editing system
4. Internal examination system
5. e-learning management system

Savitribai Phule Pune University
S.Y.B.Sc. (Computer Science)
Semester – II
Computer Science Paper - I
Course Code: CS 241
Title : Data Structures and Algorithms -II

Teaching Scheme	No. of Credits	Examination Scheme
3 Lectures / week (50 mins. duration)	02	IE : 15 marks UE: 35 marks

Prerequisites :
- Knowledge of C Programming Language
- Basic knowledge of algorithms
- Basic knowledge of linear data structures

Course Objectives
- To learn the systematic way of solving problems
- To design algorithms
- To understand the different methods of organizing large amount of data
- To efficiently implement the non-linear data structures

Course Outcomes: On completion of this course students will be able to
- Implementation of different data structures efficiently
- Usage of well-organized data structures to handle large amount of data
- Usage of appropriate data structures for problem solving

Course Contents

Chapter 1	**Tree**	**10 lectures**

1.1 Concept and Terminologies
1.2 Types of Binary trees - Binary tree, skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree, Heap
1.3 Representation – Static and Dynamic
1.4 Implementation and Operations on Binary Search Tree - Create, Insert, Delete, Search, Tree traversals– preorder, inorder, postorder (recursive implementation), Level-order traversal using queue, Counting leaf, non-leaf and total nodes, Copy, Mirror.
1.5 Applications of trees
1.5.1 Heap sort with implementation
1.5.2 Introduction to Greedy strategy, Huffman encoding (implementation using priority queue)

Chapter 2	**Efficient Search Trees**	**8 lectures**

2.1 Terminology: Balanced trees - AVL Trees, Red Black tree, splay tree, Lexical search tree -Trie
2.2 AVL Tree- concept and rotations
2.3 Red Black trees - concept, insertion and deletion.
2.4 Multi-way search tree - B and B+ tree - Insertion, Deletion

Chapter 3	**Graph**	**12 lectures**

3.1 Concept and terminologies
3.2 Graph Representation –Adjacency matrix, Adjacency list, Inverse Adjacency list, Adjacency multilist
3.3 Graph Traversals – Breadth First Search and Depth First Search

3.4 Applications of graph
3.4.1 Topological sorting
3.4.2 Use of Greedy Strategy in Minimal Spanning Trees (Prims and Kruskals algorithm)
3.4.3 Single source shortest path - Dijkstra's algorithm
3.4.4 Dynamic programming strategy, All pairs shortest path - Floyd Warshall algorithm
3.4.5 Use of graphs in social networks

Chapter 4	**Hash Table**	**6 lectures**

4.1 Concept of hashing
4.2 Terminologies – Hash table, Hash function, Bucket, Hash address, collision, synonym, overflow etc.
4.3 Properties of good hash function
4.4 Hash functions : division function, MID square , folding methods
4.5 Collision resolution techniques
4.5.1 Open Addressing - Linear probing, quadratic probing, rehashing
4.5.2 Chaining - Coalesced , separate chaining

Reference Books:

7. Fundamentals of Data Structures in C- Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, 2nd Edition, Universities Press.
8. Data Structures using C and C++-Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
9. Classic Data Structures-D. Samanta, Prentice Hall India Pvt. Ltd.
10. Data Structures: A Pseudo code approach with C, Richard Gilberg, Behrouz A. Forouzan, Cengage Learning.
11. Introduction to Algorithms—Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein--MIT Press
12. Fundamentals of Computer Algorithms-- Ellis Horowitz, SartajSahni, Sanguthevar Rajasekaran, Universities Press
13. The Algorithm Design Manual - Steven S Skiena, Springer
14. Data Structures and Program Design in C- 2nd Edition, Robert L Kruse, Bruce P. Leung, Clovis L. Tondo, Pearson Education.

Savitribai Phule Pune University
S.Y.B.Sc. (Computer Science)
Semester – II
Computer Science Paper - II
Course Code: CS 242 Title : Computer Networks-I

Teaching Scheme	No. of Credits	Examination Scheme
3 lectures / week (50 mins. duration)	**02**	IE : 15 marks UE: 35 marks

Prerequisites
Principles of Digital Electronics
Communication Principles

Course Objectives
To prepare students with basic networking concepts: data communication, protocolsand standards, various topologies and applications of network.

Course Outcomes
1. Have a good understanding of the OSI and TCP/IP Reference Models and in particular have a good knowledge of Layers.
2. Understand the working of various protocols.
3. Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies

Course Contents

Chapter 1	**Introduction to Networks and Network Models**	**4 lectures**

1.1 Data communication, components, data representation
1.2 Networks, network criteria, network types - LAN, WAN, Switching, The Internet, Accessing the Internet
1.3 Network Software- Protocol hierarchies, Design Issues of the layer, Connection Oriented and Connectionless Services,
1.4 Reference models - OSI Reference Models, TCP/IP Reference model, Connection devices in different layers, Comparison of OSI and TCP/IP Reference Models.

Chapter 2	**Lower Layers**	**10 lectures**

2.1 Communication at the physical layer, data rate limits - Noiseless channel (Nyquist bit rate), noisy channel (Shannon capacity), Performance - bandwidth, throughput, latency, bandwidth-delay product, jitter
2.2 Design issues of Data Link Layer, Services - Framing, flow control, error control, congestion control, Link layer addressing
2.3 Framing Methods - Character Count, Flag bytes with Byte Stuffing, Flags bits with Bit Stuffing, Physical Layer Coding Violations
2.4 The Channel allocation problem, Static and dynamic allocation, Media Access Methods - Taxonomy of multiple-access protocols
2.5 Switching and TCP/IP layers, Types - circuit switching, packet switching and message switching
2.6 Wired LANs - Standard Ethernet characteristics, Addressing, Access method, implementation, Fast and Gigabit Ethernet

2.7 Wireless LANs - Architectural comparison, Characteristics, Access control, IEEE 802.11 architecture, Physical layer, MAC sublayer, Bluetooth architecture, Layers

Chapter 3	**Network Layer**	**12 lectures**

3.1 Network layer services - Packetizing, Routing and forwarding, other services
3.2 Open and closed loop congestion control
3.3 IPv4 addressing- Address space, classful addressing, Subnetting, Supernetting, classless addressing, Network address resolution (NAT)
3.4 Forwarding of IP packets- based on destination address, based on label
3.5 Network Layer Protocols- Internet Protocol (IP), IPv4 datagram format, Fragmentation, options
3.6 Mobile IP-addressing, agents, Three phases
3.7 Next Generation IP- IPv6 address representation, address space, address types, IPv6 protocol, packet format, extension header, Difference between IPv4 and IPv6
3.8 Routing - General idea, Algorithms - Distance vector routing, link state routing, path-vector routing

Chapter 4	**Transport Layer**	**10 Lectures**

4.1 Transport layer Services- Process-to-process communication, Addressing, Encapsulation and decapsulation, Multiplexing and demultiplexing, Flow control, Pushing or pulling, Flow control, Buffers, Sequence numbers, Acknowledgements, sliding window, congestion control
4.2 Connectionless and Connection-oriented service, Port numbers
4.3 Transport layer protocols- User datagram protocol, user datagram, UDP services
4.4 Transmission Control Protocol - TCP Services, TCP Features, TCP Segment format, three-way handshake for connection establishment and termination, State transition diagram, windows in TCP.

Reference Books:

1. Data Communication and Networking- BehrouzFourouzan, 5th Edition, McGraw Hill Pvt. Ltd.
2. Computer Networks-Andrew S. Tanenbaum, 5th Edition, Pearson Education

Savitribai Phule Pune University
S.Y.B.Sc. (Computer Science)
Semester – II
Computer Science Paper - III
Course Code: CS 243
Title : Practical course on CS 241(Data Structures and Algorithms II) and CS 242 (Computer Networks I)

Teaching Scheme	No. of Credits	Examination Scheme
4 hrs 20 mins / week Batch size : 12	2	IE : 15 marks UE: 35 marks

Lab Book:

The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Programming Assignments:

Programs should be done individually by the student in the respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

Assessment:

Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include-timely completion, performance, innovation, efficient codes and good programming practices.

- **Internal Evaluation :**
  - 10 marks will be given based on Networking assignments.
  - 5 marks will be allocated for Assignment completion and practical attendance
- **University Evaluation :**
  - The Practical slip will be of 35 Marks which will be based on Advanced Data structures.

Operating Environment:

For Data Structures:

- **Operating system:** Linux
- **Editor:** Any linux based editor like vi, gedit etc.
- **Compiler** : cc or gcc

Course Contents :-

Assignment 1 Binary Search Tree and Traversals

1. Implement Binary Search Tree (BST) to perform following operations on BST– Create, Recursive Traversals - Inorder, Preorder, Postorder
2. Perform following operations: insert, delete

Assignment 2 Binary Search Tree Operations

1. Implement Binary Search Tree (BST) to perform following operations on BST– copy and mirror image of BST, counting leaf, non-leaf and total nodes.
2. Level-order traversal of binary search tree using queue.

Assignment 3 Applications of Binary Tree

1. Sort set of elements using Heap sort
2. Encode a set of characters using Huffman encoding

Assignment 4 Graph implementation

1. Implement Graph as adjacency matrix and adjacency list
2. Calculate indegree and outdegree of vertices
3. Graph traversals: BFS and DFS.

Assignment 5 Graph Applications

1. Implementation of Topological sorting

Assignment 6 Spanning Trees

1. Implementation of Prims/Kruskals Minimum spanning tree algorithm

Assignment 7 Shortest Path algorithms

1. Implementation of Dijkstra's shortest path algorithm for finding Shortest Path from a given source vertex using adjacency cost matrix.
2. Implementation of Floyd Warshall algorithm for all pairs shortest path.

Assignment 8 Hash Table

1. Implementation of static hash table with Linear Probing.
2. Implementation of static hash table with chaining.

Assignment 9 Hash Table-II

1. Implementation of linked hash table with chaining.

Assignment 10 Networking Assignment

Assignment 11 Networking Assignment