SavitribaiPhule Pune University, Pune
Second Year Information Technology (2019 Course)
214443:Data Structure & Algorithms

Teaching Scheme:	Credit	Examination Scheme:
TH: 03hr/week	03	Mid Semester: 30Marks End Semester: 70Marks

Prerequisite Courses, if any: Fundamental knowledge of programming language and basics of algorithms

Companion Course, if any: Discrete Structures/Discrete Mathematics

Course Objectives:

1.To study data structures and their implementations and applications.
2.To learn different searching and sorting techniques
3.To study some advanced data structures such as trees, graphs and tables.
4.To learn different file organizations.
5.To learn algorithm development and analysis of algorithms.

Course Outcomes:

On completion of the course, students will be able to–

CO1: Analyze algorithms and to determine algorithm correctness and time efficiency class.
CO2: Implement abstract data type (ADT) and data structures for given application
CO3: Design algorithms based on techniques like brute -force, divide and conquer, greedy, etc.)
CO4: Solve problems using algorithmic design techniques and data structures
CO5: Analyze of algorithms with respect to time and space complexity

COURSE CONTENT

Unit- I	Introduction	(06 Hrs)

Introduction to Data Structures: Concept of data, Data object, Data structure, Concept of Primitive and non-primitive, linear and Nonlinear, static and dynamic, persistent and ephemeral data structures, **Definition of ADT, Array:** Single and multidimensional array address calculation, recursion.

Searching and sorting: Need of searching and sorting, Concept of internal and external sorting, sort stability, Searching methods: Linear and binary search algorithms, Fibonacci Series.

Sorting methods: Bubble, insertion, Quick, Merge, shell and comparison of all sorting methods.

Case Studies	Set Operation, String Operation

Mapping of Course Outcomes for Unit I	**CO1, CO2, CO3, CO5**	
Unit- II	**Stack &Queue**	**(06 Hrs)**
Linked Organization: Concept of linked organization, Singly Linked List, Doubly Linked List, Circular Linked List as an ADT (Operations: Create, Display, Search, Insert, Delete). **Stack:** Concept of stack, Concept of implicit and explicit stack, stack as an ADT using sequential and linked organization, Applications of stack: converting expressions from infix to postfix or prefix form, evaluating postfix or prefix form. **Queue:** Concept of queues as ADT, Implementation of queue using array and linked organization, Concept of circular queue, double ended queue, Applications of queue: priority queue.		
Case Studies	**Reversing a string, balanced parentheses in algebraic expressions, Towers of Hanoi problem, double ended queue as Stack and Queue.**	
Mapping of Course Outcomes for Unit II	**CO1, CO2, CO3, CO5**	
Unit- III	**Trees**	**(06 Hrs)**
Tree : Trees and binary trees-concept and terminology, Expression tree, Binary tree as an ADT, Recursive and Non recursive algorithms for binary tree traversals, Binary search trees, Binary search tree as ADT. **Threaded binary tree:** Concept of threaded binary tree (inorder, preorder and postorder). Preorder and In-order traversals of in-order threaded binary tree, Applications of trees.		
Case Studies	**Construction of BST from pre and postorder traversal, Expression Tree construction**	
Mapping of Course Outcomes for Unit III	**CO1, CO2, CO3,CO5**	
Unit- IV	**Graph**	**(06 Hrs)**
Graph -Concept and terminologies, Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Breadth First Search traversal, Depth First Search traversal, Prim's and Kruskal's algorithms for minimum spanning tree, Shortest path using Warshall's algorithm, Shortest path using Dijkstra's algorithm, topological sorting.		
Case Studies	**Consider a network of computers connected to each other. The connection has various parameters associated with it as distance, propagation delay, bandwidth (capacity of carrying data), etc. Based on these parameters, decide which path should be chosen to send data from one computer to every other on the network.**	

Mapping of Course Outcomes for Unit IV	**CO1, CO2, CO3, CO5**	
Unit- V	**Symbol Table &Heap**	**(06 Hrs)**
Symbol Table: Notion of Symbol Table, OBST, AVL Trees Heap: Heap data structure, Min and Max Heap, Heap sort implementation, applications of heap **Hashing:** Hash tables and scattered tables: Basic concepts, hash function, characteristics of good hash function, Different key-to-address transformations techniques, synonyms or collisions, collision resolution techniques- linear probing, quadratic probing, rehashing, chaining with and without replacement.		
Case Studies	**In a system, jobs are submitted for execution at different times. If the system is idle, the job is taken for execution immediately. If there is a job in execution, the newly submitted job has to be put in a queue. The jobs are assigned a number which tells the priority of the jobs. The system must take high priority jobs first for execution. Implement the above said system using heap data structure.**	
Mapping of Course Outcomes for Unit V	**CO1, CO2, CO4**	
Unit- VI	**Analysis of Algorithms &File Organization**	**(06 Hrs)**
Analysis of algorithm: Frequency count and its importance in analysis of an algorithm,Time complexity & Space complexity of an algorithm Big 'O', 'Ω' and 'Θ' notations,Analyze Insertion sort, Quick Sort, binary search, hashing for Best, Worst and Average case. **File :**Concept of File, File types and file organization (sequential, index sequential and DirectAccess), Comparison of different file organizations.		
Case Studies	**Best case, Average case and Worst case analysis of Merge and Quick sort.**	
Mapping of Course Outcomes for Unit VI	**CO1, CO3,CO4,CO5**	

Text Books:

1. R. Gilberg, B. Forouzan, "Data Structures: A pseudo Code Approach with C++", Cengage Learning, ISBN 9788131503140.
2. E. Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN 16782928

Reference Books:

1. Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition
2. G. A.V, PAI , “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6
3. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
4. A. Tharp ,"File Organization and Processing", 2008 ,Willey India edition, 9788126518685
5. J. Tremblay, P. Soresan, "An Introduction to Data Structures with Applications", 2nd edition, Tata McGraw Hill International Editions, 1984, ISBN-0-07-462471-7.
6. M. Folk, B. Zoellick, G. Riccardi, "File Structure An Object Oriented Approach with C++", Pearson Education, 2002, ISBN 81 - 7808 - 131 - 8.
7. M. Welss, “Data Structures and Algorithm Analysis in C++", 2nd edition, Pearson Education, 2002, ISBN-81-7808-670-0

The CO-PO mapping for the course

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	3	-	3	-	-	-	-	-	2
CO2	1	3	3	3	-	3	-	-	-	-	-	2
CO3	2	1	2	3	-	3	-	-	-	-	-	2
CO4	2	3	3	3	-	3	-	-	-	-	-	2
CO5	3	3	2	3	-	3	-	-	-	-	-	2

SavitribaiPhule Pune University, Pune
Second Year Information Technology (2019 Course)
214447: Data Structure & Algorithms Lab

Teaching Scheme	Credit	Examination Scheme
PR : 04 hr/week	02	TW: 25 Marks PR: 25Marks

Prerequisite Courses, if any: Fundamental knowledge of programming language and basics of algorithms

Course Objectives:

1. To study data structures and their implementations and applications.
2. To learn different searching and sorting techniques.
3. To study some advanced data structures such as trees, graphs and tables.
4. To learn different file organizations.
5. To learn algorithm development and analysis of algorithms.

Course Outcomes:

On completion of the course, students will be able to–

CO1: Analyze algorithms and to determine algorithm correctness and time efficiency class.

CO2: Implement abstract data type (ADT) and data structures for given application

CO3: Design algorithms based on techniques like brute -force, divide and conquer, greedy, etc.)

CO4:Solve problems using algorithmic design techniques and data structures

CO5:Analyze of algorithms with respect to time and space complexity

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant.

The instructor's manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references.Experiments to be conducted in C++.

Guidelines for Student's Lab Journal

1. The laboratory assignments are to be submitted by students in the form of journals. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept, algorithms, printouts of the code written using coding

standards, sample test cases etc.

2. Practical Examination will be based on the term work.
3. Candidate is expected to know the theory involved in the experiment.
4. The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department.
5. All the assignment mentioned in the syllabus must be conducted.

Guidelines for Lab /TW Assessment

6. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
7. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.
8. Appropriate knowledge of usage of software and hardware such as compiler, debugger, coding standards, algorithm to be implemented etc. should be checked by the concerned faculty member(s).

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.

The guidelines published by BoS-IT time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory.All the assignments should be conducted on multicore hardware and 64-bit open-source software.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student ‘s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

List of Assignments

1. Searching and Sorting – CO1, CO2, CO3, CO5

Consider a student database of SEIT class(at least 15 records). Database contains different fields of every student like Roll No, Name and SGPA.(array of structure)

a. Design a roll call list, arrange list of students according to roll numbers in ascending order (Use Bubble Sort)
b. Arrange list of students alphabetically. (Use Insertion sort)
c. Arrange list of students to find out first ten toppers from a class. (Use Quick sort)
d. Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA.
e. Search a particular student according to name using binary search without recursion. (all the student records having the presence of search key should be displayed)

(Note: Implement either Bubble sort or Insertion Sort.)

2. Stack – CO1, CO2, CO3, CO5

Implement stack as an abstract data type using singly linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix and prefix expression.

3. Circular Queue – CO1, CO2, CO3, CO5

Implement Circular Queue using Linked List. Perform following operations on it.

a) Insertion (Enqueue)
b) Deletion (Dequeue)
c) Display (forward and reverse)

4. Expression Tree – CO1, CO2, CO3, CO5

Construct an Expression Tree from postfix and prefix expression. Perform recursive and non- recursive In-order, pre-order and post-order traversals.

5. Binary Search Tree – CO1, CO2, CO3, CO5

Implement binary search tree and perform following operations:

a) Insert (Handle insertion of duplicate entry)
b) Delete
c) Search
d) Display tree (Traversal)
e) Display - Depth of tree
f) Display - Mirror image
g) Create a copy

h) Display all parent nodes with their child nodes
i) Display leaf nodes
j) Display tree level wise

(Note: Insertion, Deletion, Search and Traversal are compulsory, from rest of operations, perform Any three)

6. Threaded Binary Tree – CO1, CO2, CO3, CO5

Implement In-order Threaded Binary Tree. Traverse the implemented treein Pre-order too.

7. Graph: Minimum Spanning Tree – CO1, CO2, CO3, CO5

Represent a graph of your college campus using adjacency list /adjacency matrix. Nodes should represent the various departments/institutes and links should represent the distance between them. Find minimum spanning tree using

a) Using Kruskal's algorithm.
b) Using Prim's algorithm.

Analyze above two algorithms for space and time complexity.

8. Graph: Shortest Path Algorithm – CO1, CO2, CO3, CO5

Represent a graph of city using adjacency matrix /adjacency list. Nodes should represent the various landmarks and links should represent the distance between them. Find the shortest path using Dijkstra's algorithm from single source to all destination.Analyze the implemented algorithm for space and time complexity.

9. Heap Sort - – CO1, CO2, CO4

Implement Heap sort to sort given set of values using max or min heap.

10. FILE Handling – CO1, CO3, CO5

Department maintains student's database. The file contains roll number, name, division and address. Write a program to create a sequential file to store and maintain student data. It should allow the user to add, delete information of student. Display information of particular student. If record of student does not exist an appropriate message is displayed. If student record is found it should display the student details.

Text Books

1. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures: A Pseudocode Approach using C++", Cengage Learning, 5th Edition, ISBN 978-8131504925
2. Mark Allen Weiss, "Data structures and Algorithm Analysis in C++ ", Pearson Education India, 3 edition (2007), ISBN 978-8131714744
3. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, "Fundamentals of Data Structures in C++", University Press (2008), ISBN 978-8173716065

Reference Books

1. Hemant Jain, "Problem Solving in Data Structures & Algorithms using C++", CreateSpace Independent Publishing Platform (2017), ISBN 978-1542396479.
2. G A V PAI, "DATA STRUCTURES and Algorithms Concepts, Techniques and Applications", McGraw Hill (2017), ISBN 978-0070667266
3. Michael T. Goodrich, Roberto Tamassia, David Mount, “Data Structures and Algorithms in C++ “, Wiley (2007), ISBN 978-8126512607
4. E Balagurusamy, "Object-Oriented Programming with C++", McGraw Hill Education; Seventh edition (2017), ISBN 978-9352607990.

The CO-PO mapping for the course

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	3	-	3	-	-	-	-	-	2
CO2	1	3	3	3	-	3	-	-	-	-	-	2
CO3	2	1	2	3	-	3	-	-	-	-	-	2
CO4	2	3	3	3	-	3	-	-	-	-	-	2
CO5	3	3	2	3	-	3	-	-	-	-	-	2