



Savitribai Phule Pune University, Pune Second Year Artificial Intelligence & Machine Learning (2020 Course) 218546 : Data Structure & Algorithms Laboratory		
Teaching Scheme:	Credit Scheme:	Examination Scheme:
Practical (PR) : 04 hrs/week	02	PR : 25 Marks TW: 25 Marks
Prerequisite Courses, if any: Fundamental knowledge of programming language and basics of algorithms		
Course Objectives: <ol style="list-style-type: none"> 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		
<ol style="list-style-type: none"> 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
<ol style="list-style-type: none"> 1. 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. 2. 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. 3. 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
<p>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.</p> <p>All the assignments should be conducted on multicore hardware and 64-bit open-source software.</p>
Guidelines for Practical Examination
<p>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.</p>
List of Assignments
Virtual Laboratory
<ul style="list-style-type: none"> • https://ds1-iiith.vlabs.ac.in/data-structures-1/ • https://ds2-iiith.vlabs.ac.in/data-structures-2/ • http://cse01-iiith.vlabs.ac.in/
1. Searching and Sorting -- CO1, CO2, CO3, CO5
<p>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)</p> <ol style="list-style-type: none"> 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

<p>student records having the presence of search key should be displayed) (Note: Implement either Bubble sort or Insertion Sort.)</p>
<p>2. Stack -- CO1, CO2, CO3, CO5</p>
<p>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.</p>
<p>3. Circular Queue -- CO1, CO2, CO3, CO5</p>
<p>Implement Circular Queue using Array. Perform following operations on it.</p> <ol style="list-style-type: none"> Insertion (Enqueue) Deletion (Dequeue) Display <p>(Note: Handle queue full condition by considering a fixed size of a queue.)</p>
<p>4. Expression Tree -- CO1, CO2, CO3, CO5</p>
<p>Construct an Expression Tree from postfix and prefix expression. Perform recursive and non- recursive In-order, pre-order and post-order traversals.</p>
<p>5. Binary Search Tree -- CO1, CO2, CO3, CO5</p>
<p>Implement binary search tree and perform following operations:</p> <ol style="list-style-type: none"> Insert (Handle insertion of duplicate entry) Delete Search Display tree (Traversal) Display - Depth of tree Display - Mirror image Create a copy Display all parent nodes with their child nodes Display leaf nodes Display tree level wise <p>(Note: Insertion, Deletion, Search and Traversal are compulsory, from rest of operations, perform Any three)</p>
<p>6. Threaded Binary Tree -- CO1, CO2, CO3, CO5</p>
<p>Implement In-order Threaded Binary Tree and traverse it in In-order and Pre-order.</p>
<p>7. Graph: Minimum Spanning Tree -- CO1, CO2, CO3, CO5</p>
<p>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</p> <ol style="list-style-type: none"> Using Kruskal's algorithm. Using Prim's algorithm.
<p>8. Graph: Shortest Path Algorithm -- CO1, CO2, CO3, CO5</p>

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.

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