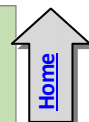




**Savitribai Phule Pune University**  
**Second Year of Computer Engineering (2019 Course)**  
**210256: Data Structures and Algorithms Laboratory**



<b>Teaching Scheme</b> <b>Practical: 04 Hours/Week</b>	<b>Credit Scheme</b> <b>02</b>	<b>Examination Scheme and Marks</b> <b>Term Work: 25 Marks</b> <b>Practical: 25 Marks</b>
---	-----------------------------------	---

**Companion Course :** 210252: Data Structures and Algorithms, 210255: Principles of Programming Languages

**Course Objectives:**

- To **understand** practical implementation and usage of non linear data structures for solving problems of different domain.
- To strengthen the ability to identify and **apply** the suitable data structure for the given real world problems.
- To **analyze** advanced data structures including hash table, dictionary, trees, graphs, sorting algorithms and file organization.

**Course Outcomes:**

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

- CO1: Understand** the ADT/libraries, hash tables and dictionary to design algorithms for a specific problem.
- CO2:** Choose most appropriate data structures and **apply** algorithms for graphical solutions of the problems.
- CO3: Apply** and **analyze** non linear data structures to solve real world complex problems.
- CO4: Apply** and **analyze** algorithm design techniques for indexing, sorting, multi-way searching, file organization and compression.
- CO5: Analyze** the efficiency of most appropriate data structure for creating efficient solutions for engineering design situations.

**Guidelines for Instructor's Manual**

The instructor's manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of course, conduction and Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

**Guidelines for Student's Laboratory Journal**

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

**Guidelines for Laboratory / Term Work Assessment**

Continuous assessment of laboratory work should be done based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

**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 assignment framing policy need to address the

average students and inclusive of an element to attract and promote the intelligent students. 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. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged.

In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Set of suggested assignment list is provided in groups- A, B, C, D, E, F and G. Each student must perform at least 12 assignments( at least 02 from group A, 03 from group B, 02 from group C, 2 from group D, 01 from group E, 02 from group F.)

**Operating System recommended** :- 64-bit Open source Linux or its derivative

**Programming tools recommended** : - Open Source Python - Group A assignments, C++ Programming tool like G++/GCC

### Guidelines for Practical Examination

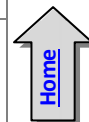
Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. Consequently encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. Therefore adhering to these principles will consummate our team efforts to the promising start of the student's academics.

### Virtual Laboratory:

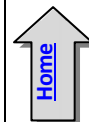
- <http://cse01-iiith.vlabs.ac.in/Courses%20Aligned.html?domain=Computer%20Science>

### Suggested List of Laboratory Experiments/Assignments

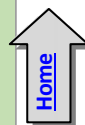
Sr. No	Group A
1	Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers
2	Implement all the functions of a dictionary (ADT) using hashing and handle collisions using chaining with / without replacement. Data: Set of (key, value) pairs, Keys are mapped to values, Keys must be comparable, Keys must be unique. Standard Operations: Insert(key, value), Find(key), Delete(key)
3	For given set of elements create skip list. Find the element in the set that is closest to some given value. (note: Decide the level of element in the list Randomly with some upper limit)
4	To create ADT that implement the "set" concept. a. Add (new Element) -Place a value into the set , b. Remove (element) Remove the value c. Contains (element) Return true if element is in collection, d. Size () Return number of values in collection Iterator () Return an iterator used to loop over collection, e. Intersection of two sets , f. Union of two sets, g. Difference between two sets, h. Subset
	Group B
5	A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.
6	Beginning with an empty binary search tree, Construct binary search tree by inserting the values in the order given. After constructing a binary tree - i. Insert new node, ii. Find number of nodes in longest path from root, iii. Minimum data value found in the tree, iv. Change a tree so that the roles of the left and right pointers are swapped at every node, v. Search a value



7	Construct an expression tree from the given prefix expression eg. $+-a*bc/def$ and traverse it using post order traversal (non recursive) and then delete the entire tree.
8	Read for the formulas in propositional calculus. Write a function that reads such a formula and creates its binary tree representation. What is the complexity of your function?
9	Convert given binary tree into threaded binary tree. Analyze time and space complexity of the algorithm.
10	Consider threading a binary tree using preorder threads rather than inorder threads. Design an algorithm for traversal without using stack and analyze its complexity. _
11	A Dictionary stores keywords and its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Binary Search Tree for implementation.
12	Implement a file compression algorithm that uses binary tree. Your program should allow the user to compress and decompress messages containing alphabets using the standard Huffman algorithm for encoding and decoding.
<b>Group C</b>	
13	Represent a given graph using adjacency matrix/list to perform DFS and using adjacency list to perform BFS. Use the map of the area around the college as the graph. Identify the prominent land marks as nodes and perform DFS and BFS on that.
14	There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight take to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Check whether the graph is connected or not. Justify the storage representation used.
15	You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.
16	Tour operator organizes guided bus trips across the Maharashtra. Tourists may have different preferences. Tour operator offers a choice from many different routes. Every day the bus moves from starting city S to another city F as chosen by client. On this way, the tourists can see the sights alongside the route travelled from S to F. Client may have preference to choose route. There is a restriction on the routes that the tourists may choose from, the bus has to take a short route from S to F or a route having one distance unit longer than the minimal distance. Two routes from S to F are considered different if there is at least one road from a city A to a city B which is part of one route, but not of the other route.
17	Consider the scheduling problem. n tasks to be scheduled on single processor. Let $t_1, \dots, t_n$ be durations required to execute on single processor is known. The tasks can be executed in any order but one task at a time. Design a greedy algorithm for this problem and find a schedule that minimizes the total time spent by all the tasks in the system. (The time spent by one is the sum of the waiting time of task and the time spent on its execution.)
<b>Group D</b>	
18	Given sequence $k = k_1 < k_2 < \dots < k_n$ of n sorted keys, with a search probability $p_i$ for each key $k_i$ . Build the Binary search tree that has the least search cost given the access probability for each key?



19	A Dictionary stores keywords and its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword											
	Group E											
20	Consider a scenario for Hospital to cater services to different kinds of patients as Serious (top priority), b) non-serious (medium priority), c) General Checkup (Least priority) Implement the priority queue to cater services to the patients.											
21	Implement the Heap/Shell sort algorithm implemented in Java demonstrating heap/shell data structure with modularity of programming language											
22	Read the marks obtained by students of second year in an online examination of particular subject. Find out maximum and minimum marks obtained in that subject. Use heap data structure. Analyze the algorithm.											
	Group F											
23	Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data.											
24	Company maintains employee information as employee ID, name, designation and salary. Allow user to add, delete information of employee. Display information of particular employee. If employee does not exist an appropriate message is displayed. If it is, then the system displays the employee details. Use index sequential file to maintain the data.											
25	Implementation of a direct access file -Insertion and deletion of a record from a direct access file											
26	Assume we have two input and two output tapes to perform the sorting. The internal memory can hold and sort m records at a time. Write a program in java for external sorting. Find out time complexity.											
	Mini-Projects/ Case Study											
27	Design a mini project using JAVA which will use the different data structure with or without Java collection library and show the use of specific data structure on the efficiency (performance) of the code.											
28	Design a mini project to implement Snake and Ladders Game using Python.											
29	Design a mini project to implement a Smart text editor.											
30	Design a mini project for automated Term work assessment of student based on parameters like daily attendance, Unit Test / Prelim performance, Students achievements if any, Mock Practical.											
@The CO-PO Mapping Matrix												
PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	-	-	-	-	-	-	-	-	-
CO2	-	2	2	-	-	-	-	-	-	-	-	-
CO3	-	2	2	1	-	-	-	-	-	-	-	-
CO4	1	2	1	1	-	-	-	-	-	-	-	-
CO5	1	1	2	2	-	-	-	-	-	-	-	-



<b>Savitribai Phule Pune University</b> <b>Second Year of Computer Engineering (2019 Course)</b> <b>210257: Microprocessor Laboratory</b>		
<b>Teaching Scheme</b> <b>Practical: 02 Hours/Week</b>	<b>Credit Scheme</b> <b>01</b>	<b>Examination Scheme and Marks</b> <b>Term Work: 25 Marks</b> <b>Oral: 25 Marks</b>
<b>Companion Course :</b> 210254: Microprocessor		
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>To understand assembly language programming instruction set</li> <li>To understand different assembler directives with example</li> <li>To apply instruction set for implementing X86/64 bit assembly language programs</li> </ul>		
<b>Course Outcomes:</b> On completion of the course, learner will be able to– CO1. <b>Understand</b> and <b>apply</b> various addressing modes and instruction set to implement assembly language programs CO2. <b>Apply</b> logic to <b>implement</b> code conversion CO3. <b>Analyze</b> and <b>apply</b> logic to <b>demonstrate</b> processor mode of operation		
<b>Guidelines for Laboratory /Term Work Assessment</b> Continuous assessment of laboratory work is based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.		
<b>Guidelines for Laboratory Conduction</b> The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. 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. Use of open source software is encouraged. In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus. Operating System: 64-bit Open source Linux or its derivative. Programming Tools: Preferably using Linux equivalent or MASM/TASM/NASM/FASM.		
<b>Guidelines for Practical Examination</b> Both internal and external examiners should jointly set problem statements. 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 test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.		
<b>Virtual Laboratory:</b> <ul style="list-style-type: none"> <li><a href="http://209.211.220.205/vlabiitece/mi/MI3.php">http://209.211.220.205/vlabiitece/mi/MI3.php</a></li> </ul>		
<b>Suggested List of Laboratory Experiments/Assignments(any 10)</b>		
<b>Sr. No.</b>	<b>Assignments</b>	

1	Write an X86/64 ALP to accept five 64 bit Hexadecimal numbers from user and store them in an array and display the accepted numbers.
2	Write an X86/64 ALP to accept a string and to display its length.
3	Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers.
4	Write a switch case driven X86/64 ALP to perform 64-bit hexadecimal arithmetic operations (+, -, *, /) using suitable macros. Define procedure for each operation.
5	Write an X86/64 ALP to count number of positive and negative numbers from the array.
6	Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. (Wherever necessary, use 64-bit registers).
7	Write X86/64 ALP to detect protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers also identify CPU type using CPUID instruction.
8	Write X86/64 ALP to perform non-overlapped block transfer without string specific instructions. Block containing data can be defined in the data segment.
9	Write X86/64 ALP to perform overlapped block transfer with string specific instructions. Block containing data can be defined in the data segment.
10	Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. (use of 64-bit registers is expected).
11	Write X86 Assembly Language Program (ALP) to implement following OS commands i) COPY, ii) TYPE Using file operations. User is supposed to provide command line arguments
12	Write X86 ALP to find, a) Number of Blank spaces b) Number of lines c) Occurrence of a particular character. Accept the data from the text file. The text file has to be accessed during Program_1 execution and write FAR PROCEDURES in Program_2 for the rest of the processing. Use of PUBLIC and EXTERN directives is mandatory.
13	Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code.
14	Write an X86/64 ALP password program that operates as follows: a. Do not display what is actually typed instead display asterisk ("*"). If the password is correct display, "access is granted" else display "Access not Granted"
15	<b>Study Assignment:</b> Motherboards are complex. Break them down, component by component, and Understand how they work. Choosing a motherboard is a hugely important part of building a PC. Study- Block diagram, Processor Socket, Expansion Slots, SATA, RAM, Form Factor, BIOS, Internal Connectors, External Ports, Peripherals and Data Transfer, Display, Audio, Networking, Overclocking, and Cooling. 4. <a href="https://www.intel.in/content/www/in/en/support/articles/000006014/boards-and-kits/desktop-boards.html">https://www.intel.in/content/www/in/en/support/articles/000006014/boards-and-kits/desktop-boards.html</a>

### @The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	1	2	1	-	-	-	-	-	-	-	-
CO2	2	1	-	1	-	-	-	-	-	-	-	-
CO3	-	1	-	1	-	-	-	-	-	-	-	-



