

# PUNE INSTITUTE OF COMPUTER TECHNOLOGY

## INFORMATION TECHNOLOGY

ACADEMIC YEAR -2020\_21

Semester – II: Subject: DSAL Lab

### ASSIGNMENT1 WRITEUP FORMAT

1.	Title	Assignment 1: Searching and Sorting
2.	Aim	<b>To implement Searching and Sorting on Array as a linear Data structure</b>
3.	Problem statement	<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> <p>a. Design a roll call list, arrange list of students according to roll numbers in ascending order (Use Bubble Sort)</p> <p>b. Arrange list of students alphabetically. (Use Insertion sort)</p> <p>c. Arrange list of students to find out first ten toppers from a class. (Use Quick sort)</p> <p>d. Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA.</p> <p>e. Search a particular student according to name using binary search without recursion.</p> <p>(all the student records having the presence of search key should be displayed) (Note: Implement either Bubble sort or Insertion Sort.)</p>
4.	Objective	<ul style="list-style-type: none"> <li>To study the concepts of array of structure.</li> <li>Know the concepts, algorithms and applications of Sorting.</li> <li>Understand the concepts, algorithms and applications of Searching.</li> </ul>
5.	Outcome	<ul style="list-style-type: none"> <li>Understand linear data structure - array of structure.</li> <li>Apply different sorting techniques on array of structure (Bubble, Insertion and Quicksort) and display the output for every pass.</li> <li>Apply different searching techniques on array of structure (Linear Search, Binary Search) and display the output for every pass..</li> <li>Calculate time complexity.</li> </ul>
6.	Theory	<p><b>Theory ::</b></p> <ol style="list-style-type: none"> <li>Concept of Linear data structure</li> <li>Example of Linear data structure</li> <li>Array:: (1 D Array) <ol style="list-style-type: none"> <li>Concepts</li> <li>Definition of Array</li> <li>Need of Array</li> </ol> </li> <li>ADT of Array <ol style="list-style-type: none"> <li>Write proper ADT format (ADT should not include any implementation concept)</li> </ol> </li> <li>Graphical representation of an array (show following points) <ol style="list-style-type: none"> <li>Memory Map of Array</li> <li>Terminology with Array (base address, constant pointers, Index, Subscript)</li> <li>Address calculations in Array</li> </ol> </li> <li>Array and structure in C++</li> <li>Declaration, initialization, input output of array in C++</li> <li>Applications of array</li> <li>Basics of structure data types in C++</li> <li>Array of structures in C++ declaration, definition, input, output</li> <li>Assignment one database definition of your choice</li> </ol>

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

INFORMATION TECHNOLOGY

ACADEMIC YEAR -2020\_21

Semester – II: Subject: DSAL Lab

ASSIGNMENT1 WRITEUP FORMAT

		12. Basics of searching and sorting 13. Application of searching and sorting
7.	<b>Algorithms /Pseudocode:</b>	1. Explain linear search with example and pseudocode 2. Explain Binary search with example and pseudocode 3. Explain Bubble sort with example and pseudocode 4. Explain Insertion Sort with example and pseudocode 5. Explain Quick Sort with example and pseudocode
8.	Test cases/validation	<b>Validations :</b> <ul style="list-style-type: none"> <li>Limit of the array should not be -ve, and should not cross the lower and upper bound.</li> <li>Roll numbers should not repeat, should not -ve</li> <li>Name should only contain alphabets, space and .</li> <li>Should not allowed any other operations before the input list is entered by the user .</li> <li>Before going to (binary search) records should be sorted according a names.</li> </ul> <b>Test Cases</b> Sorting : <ol style="list-style-type: none"> <li>Four test cases :</li> <li>Already Sorted according to the requirement</li> <li>Sorted in reverse order</li> <li>Partially sorted</li> <li>Completely Random List</li> </ol> →Show pass wise output , including no of comparisons and swaps needed Searching : <ol style="list-style-type: none"> <li>Search element at first, middle, last position.</li> <li>Search element is not in the list</li> </ol>
9.	Program	<b>Printout /Softcopy</b>
10.	Results /output	<b>Including test cases , validations and valid inputs based results .</b>
11	Conclusion	Add the comparative study and analysis in tabular format <ol style="list-style-type: none"> <li>All sorting algorithms: consider following factors for comparison : Time complexity : best, average worst , Space complexity, stable sort or not, methods used for sorting</li> <li>Searching algorithms: Consider following factors for comparison: <ol style="list-style-type: none"> <li>Time complexity : best, average, worst</li> <li>Input list constraint</li> <li>Methods used for sorting.</li> </ol> </li> </ol>

**Subject Coordinator**  
**S.H. Chandak**