

Branch Name:	Artificial Intelligence & Data Science
Semester/Year:	Semester - III/ Second Year
Subject Title:	Data Structures and Algorithm with Python
Subject Code:	1ET1080303
Pre-requisite:	Fundamentals of OOP

Course Objective:

The Objective of this course is to make the students learn about the mapping between data structures & algorithms, designing and analyzing algorithms for real world problems.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
4	-	2	5	70	30	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	Introduction: Elementary Data Organization: Primitive and Non-primitive data types, Data Structure Types and Operations, Abstract Data Types, Analysis of Algorithms, Space and Time Complexity. Informal introduction to programming, algorithms and data structures via gcd, Downloading and installing Python, gcd in Python: Variables, operations, control flow - assignments, condition- als, loops, functions.	5	9
2	Array, Structure and Pointer: Representation of arrays, Applications of arrays, sparse matrix and its representation, Declaration of structures, Use of structure through function, Pointer variable, Pointer Arithmetic, Passing parameters by reference, Pointers to Functions, Dynamic memory allocation, Pointer to Structures.	3	5
3	Stack and Queue: ADT Stacks, Implementation of stack using array and Linked List, Application of stack: Conversion of Infix to Postfix, Evaluation of Postfix, ADT queue, Implementation of Queue, Circular Queue, Concept of Priority Queue, Application of Queue	6	15

4	Linked List: Singly linked list: Representation of linked list in memory, Traversing, Searching, Insertion, Deletion from linked list, Polynomial Addition, Doubly Linked list, Circular Linked List.	6	15
5	Tree: Tree Terminology, Binary Tree, Binary Tree Traversal, Conversion Of General Trees To Binary Trees, Binary Search tree: Insertion, Deletion, Traversal, AVL Trees, Heap, Multi way trees, B-Trees, B+ Tree, 2-3 tree, applications of Trees-the Manipulation of Arithmetic Expressions	5	10
6	Graphs: Matrix representation of graphs, Breadth First Search, Depth First Search, Spanning Trees	5	10
7	Hashing: Hashing: The symbol table, Hashing Functions, Collision-Resolution Techniques	2	9
8	Searching and Sorting: Linear Search, Binary Search, Selection sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Radix Sort, Shell Sort	8	12
9	File Structure: Definition of Record, File, Block, Sequential, Indexed and Relative/Random File Organization, Indexing structure for index files, hashing for direct files, Multi - Key file organization and access methods	5	15

Course Outcome:

At the end of this course, the student would be able

1. This course is an introduction to programming and problem solving in Python. It does not assume any prior knowledge of programming.
2. Using some motivating examples, the course quickly builds up basic concepts such as conditionals, loops, functions, lists, strings and tuples.
3. It goes on to cover searching and sorting algorithms, dynamic programming and backtracking, as well as topics such as exception handling and using flesh. As far as data structures are concerned,
4. The course covers Python dictionaries as well as classes and objects for fining user defined data types such as linked lists and binary search trees.

List of References:

1. An Introduction to Data Structures with Applications. By Jean - Paul Tremblay & Paul G.Sorenson
Publisher - Tata McGraw Hill.
2. Fundamentals of Computer Algorithms by Horowitz, Sahni, Galgotia Pub. 2001 ed.
3. Introduction to Computation and Programming Using Python by John V Guttag (PHI)
4. Core Python Programming by R. Nageswara Rao, Dreamtech
5. Core Python Programming Second Edition by Wesley J. Chun. (PHI)
6. Professional Python by Luke Sneeringer, Wrox
7. A Practical Introduction to Python Programming by Brian Heinold

e-Resource:

1. <http://nptel.ac.in/courses/106102064/1>
2. https://www.tutorialspoint.com/data_structures_algorithms/
3. <https://www.python.org/>
4. <https://nptel.ac.in/courses/117106113/34>

List of Suggested Experiments:

- 1 Write a program to check whether given matrix is magic square matrix or not.
- 2 Write a program to implement following string operation using pointers.
(a) Strlen b)Strcpy (c)Strcat (d)Strcmp(e)Strrev
- 3 Implement a program for stack that performs following operations using array.
(a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY
- 4 Implement a program to convert infix notation to postfix notation using stack.
- 5 WAP to implement QUEUE using arrays that performs following operations
(a) INSERT (b) DELETE (c) DISPLAY
- 6 Write a program to implement Circular Queue using arrays that performs following operations. (a) INSERT (b) DELETE (c) DISPLAY
- 7 Write a program for insert routine in input restricted queues.
- 8 Write a menu driven program to implement following operations on the singlylinked list.
 - (a) Insert a node at the front of the linked list.
 - (b) Insert a node at the end of the linked list.
 - (c) Insert a node such that linked list is in ascending order.
 - (d) Delete a first node of the linked list.
 - (e) Delete a node before specified position.
 - (f) Delete a node after specified position
- 9 Write a program to concatenate two doubly linked lists.
- 10 Implement recursive and non-recursive tree traversing methods in-order, preorder and post-order traversal.
- 11 Write a program to sort an array using Bubble Sort and then search an element into the array using BinarySearch.
- 12 Write a program to sort an elements using
 - (a) Insertion Sort (b) Selection Sort (c) Quick Sort (d) MergeSort