EE/EC212	DATA STRUCTURES USING PYTHON	L	T	P	C	Int	Ext
		2	1	-	3.0	30	70

Semester III [Second Year]

COURSE OBJECTIVES:

The students are able to

1. introduce object-oriented concepts in Python language.
2. familiarize linear and nonlinear data structures and their applications.
3. provide knowledge of various searching, sorting and hashing techniques.

COURSE OUTCOMES:

After successful completion of the course, the students are able to

1. define object oriented features for solving real world problems, analyze algorithms.
2. construct various linear data structures and applications.
3. apply Sorting, Searching to organize and retrieve data efficiently.
4. solve problems using non-linear data structures and Hashing techniques.

UNIT I [CO:1] **(10)**

Object-Oriented Programming: Goals, Principles and Patterns - Software Development - Class Definitions - Python's Built-In Classes, Operator Overloading and Python's Special Methods, Inheritance - Extending, Hierarchy of Numeric Progressions, Abstract Base Classes, Namespaces and Object-Orientation - Shallow and Deep Copying.

Algorithm Analysis: Asymptotic Analysis.

UNIT II [CO:2] **(10)**

Stacks: The Stack Abstract Data Type, Simple Array-Based Stack Implementation, Reversing Data Using a Stack, Matching Parentheses and HTML Tags.

Queues: The Queue Abstract Data Type, Array-Based Queue Implementation.

Double-Ended Queues: The Deque Abstract Data Type, Implementing a Deque with a Circular Array, Deques in the Python Collections Module.Priority Queues: The Priority Queue Abstract Data Type, Implementing a Priority Queue.

UNIT III [CO:2,3] **(10)**

Linked Lists: Singly Linked Lists - Circularly Linked Lists - Doubly Linked Lists - Link Based vs Array Based Sequences.

Sorting and Selection: Why Study Sorting Algorithms? - Linear Search -Binary search - Bubble sort - Selection Sort - Insertion Sort - Merge-Sort - Quick-Sort.

UNIT IV [CO:4] **(10)**

Trees: General Trees- Binary Trees - Implementing Trees - Tree Traversal Algorithms - Binary Search Trees.

Graphs: Graphs - Graph Traversals.

Hash Tables: Hash Functions - Collision Handling Schemes.

LEARNING RESOURCES:

TEXT BOOK:

Michael T.Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Python", John Wiley & Sons, 2013.

REFERENCE BOOK(s):

1. Rance D. Necaise, Data Structures and Algorithms using Python, John Wiley & Sons, 2011.
2. Narasimha Karumanchi, Data Structures and Algorithmic Thinking with Python, CareerMonk Publications, 2016.

WEB RESOURCES:

1. https://www.tutorialspoint.com/python_data_structure/index.htm
2. https://www.edureka.co/blog/data-structures-in-python/