



SRI VENKATESWARA UNIVERSITY

**4-Year UG Honours in B.Sc. (Data Mining) as Major in consonance with
Curriculum framework (W.E.F. AY-2026-27)**

COURSE STRUCTURE (for Semester III)

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
II	III	5	Data Base Management Systems	3	3
			Data Base Management Systems-Practical	2	1
		6	Data Structures & Algorithms using python	3	3
			Data Structures & Algorithms using Python-Practical	2	1
		7	Data Mining Techniques using R	3	3
			Data Mining Techniques using R -Practical	2	1



SRI VENKATESWARA UNIVERSITY

II Year Syllabus for 4-Year UG Honours in B.Sc. (Data Mining) as Major in consonance with the Curriculum framework w.e.f. AY 2026-27

II Year III Semester

Sl. No.	Category	Course No.	Course Name	No. of Hours		Total No. of Hours	No. of Credits		Total No. of Credits
				Theory	Practical		Theory	Practical	
1	Major-Core	5	Database Management Systems	3	2	5	3	1	4
2	Major-Core	6	Data Structures & Algorithms	3	2	5	3	1	4
3	Major-Core	7	Data Mining Techniques using R	3	2	5	3	1	4
4	Minor	1		3	2	5	3	1	4
5	AECC	1	English	4	0	4	3	0	3
6	AECC	1	MIL (Telugu/Hindi/Sanskrit)	4	0	4	3	0	3
7	MDC	2	-	2	0	2	2	0	2
8	SEC	3	SEC: Design Thinking/Problem solving/ others	2	0	2	2	0	2
				24	8	32	22	4	26

SEMESTER-III

COURSE 6: DATA STRUCTURES & ALGORITHMS USING PYTHON

Theory

Credits: 3

3 hrs/week

Course objectives:

1. To learn Python basics and analyze algorithms for efficiency.
2. To implement linear data structures like arrays, stacks, and queues.
3. To study non-linear structures such as linked lists, trees, and heaps.
4. To apply graph algorithms, hashing, and advanced data structures.
5. To understand searching, sorting, greedy, and dynamic programming techniques.

Syllabus:

Unit 1: Algorithm Analysis

Algorithm Definition, Characteristics, Algorithm analysis: time complexity, space complexity, Big-O, Big-Ω, Big-Θ notations, Simple examples (factorial, Fibonacci, searching minimum/maximum).

Unit 2: Linear Data Structures

Arrays and Lists in Python, Stacks: implementation using lists, Queues: simple, circular, priority queues, Deque (double-ended queue), Applications: expression evaluation, parentheses matching, job scheduling.

Unit 3: Non-Linear Data Structures

Linked Lists: singly, doubly, circular, Trees: binary tree, binary search tree (BST) Tree traversal: inorder, preorder, postorder, Applications of trees (expression trees, hierarchy representation), Heaps and Priority Queues (min-heap, max-heap).

Unit 4: Graphs and Advanced Data Structures

Graph representation: adjacency matrix, adjacency list, Graph traversal: BFS, DFS, Shortest path algorithms: Dijkstra's, Bellman-Ford, Minimum spanning tree: Kruskal's, Prim's, Hashing and hash tables.

Unit 5: Sorting, Searching, and Advanced Topics

Searching: Linear search, Binary search, Sorting: Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort, Heap sort, Comparison of sorting algorithms, Introduction to advanced topics: Dynamic Programming (Fibonacci, Knapsack), Introduction to Greedy algorithms (Activity selection, Huffman coding).

Prescribed Textbooks:

1. "Data Structures and Algorithms in Python" – Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser.
2. "Problem Solving with Algorithms and Data Structures Using Python" – Bradley N. Miller, David L. Ranum.
3. "Data Structures and Algorithms in Python" – Benjamin Baka.

SEMESTER-III

COURSE 6: DATA STRUCTURES & ALGORITHMS USING PYTHON

Practical

Credits: 1

2 hrs/week

List of experiments:

1. Write a Python program to perform array operations: insertion, deletion, and traversal.
2. Write a Python program to implement a stack using lists with push, pop, and peek operations.
3. Write a Python program to implement a queue using lists with enqueue and dequeue operations.
4. Write a Python program to implement a circular queue / deque.
5. Write a Python program to create a singly linked list with insertion and deletion operations.
6. Write a Python program to create a doubly linked list with insertion and deletion operations.
7. Write a Python program to implement tree traversals (inorder, preorder, postorder).
8. Write a Python program to implement a binary search tree with insertion and searching.
9. Write a Python program to implement a heap (min-heap and max-heap).
10. Write a Python program to implement graph traversal algorithms (BFS and DFS).
11. Write a Python program to implement searching techniques: Linear Search and Binary Search.
12. Write a Python program to implement sorting techniques: Bubble Sort, Insertion Sort, and Quick Sort.

SRI VENKATESWARA UNIVERSITY::TIRUPATI
B.Sc. Data Mining Honours
II year III Semester
COURSE 6: Data Structures & Algorithms using python

MODEL QUESTION PAPER

Time: **3 Hours**

Max. Marks: **75**

PART-A

Answer any **FIVE** of the following. Each Question Carries 3 marks.

(5X3=15)

1. Short answer question from unit-I
2. Short answer question from unit-I
3. Short answer question from unit-II
4. Short answer question from unit-II
5. Short answer question from unit-III
6. Short answer question from unit-III
7. Short answer question from unit-IV
8. Short answer question from unit-IV
9. Short answer question from unit-V
10. Short answer question from unit-V

PART-B

Answer **FIVE** of the following. Each Question Carries 12 marks.

(5X12=60)

11. a) Essay question from Unit-I
OR
b) Essay question from Unit-I
12. a) Essay question from Unit-II
OR
b) Essay question from Unit-II
13. a) Essay question from Unit-III
OR
b) Essay question from Unit-III
14. a) Essay question from Unit-IV
OR
b) Essay question from Unit-IV
15. a) Essay question from Unit-V
OR
b) Essay question from Unit-V