

**Tribhuvan University**  
**Institute of Science and Technology**  
**Department of Computer Science and Technology**

**Data Structure and Algorithms Micro-Syllabus**

S. No.	Unit	Hours	Total Hours	Marks
1	<b>Concept and definition of Data Structures</b> a. Information and its meaning b. Array in C c. The array as an ADT d. One dimensional array e. Two dimensional array f. Multi-dimensional array g. Structure h. Union i. Pointer	4	4	5
2	<b>Algorithm</b> a. Concept and definition b. Design of algorithm c. Characteristics of algorithm d. Big O notation	2	2	3
3	<b>The Stack</b> a. Concept and Definition <ul style="list-style-type: none"><li>• Primitive Operations</li><li>• Stack as an ADT</li><li>• Implementing PUSH and POP operation</li><li>• Testing for overflow and underflow conditions</li></ul> b. The Infix, Postfix and Prefix <ul style="list-style-type: none"><li>• Concept and Definition</li><li>• Evaluating the postfix operation</li><li>• Converting from infix to postfix</li></ul> c. Recursion <ul style="list-style-type: none"><li>• Concept and Definition</li><li>• Implementation of :<ul style="list-style-type: none"><li>➤ multiplication of Natural Numbers</li><li>➤ Factorial</li><li>➤ Fibonacci Sequences</li><li>➤ The Tower of Hanoi</li></ul></li></ul>	2    3   3	8	11
4	<b>Queues</b> a. Concept and Definition b. Queue as ADT c. Implementation of Insert and Delete operation of <ul style="list-style-type: none"><li>• Linear Queue</li><li>• Circular Queue</li></ul> d. Concept of Priority Queue	1  2  1	4	5
5	<b>Linked List</b> a. Concept and Definition b. Inserting and deleting nodes c. Linked implementation of a stack (PUSH/POP) d. Linked implementation of queue (Insert/Remove) e. Circular List <ul style="list-style-type: none"><li>• Stack as a circular list (PUSH / POP)</li></ul>	1 2 2	6	8

	<ul style="list-style-type: none"> <li>• Queue as a circular list (Insert / Remove)</li> </ul> f. Doubly Linked List (Insert/Remove)			
6	<b>Tree</b> <ol style="list-style-type: none"> <li>a. Concept and Definition</li> <li>b. Binary Tree</li> <li>c. Introduction and application</li> <li>d. Operation</li> <li>e. Types of Binary Tree           <ul style="list-style-type: none"> <li>• Complete</li> <li>• Strictly</li> <li>• Almost Complete</li> </ul> </li> <li>f. Huffman algorithm</li> <li>g. Binary Search Tree           <ul style="list-style-type: none"> <li>• Insertion</li> <li>• Deletion</li> <li>• Searching</li> </ul> </li> <li>h. Tree traversal           <ul style="list-style-type: none"> <li>• Pre-order transversal</li> <li>• In-order transversal</li> <li>• Post-order transversal</li> </ul> </li> </ol>	1 1 1 2 2	7	9
7	<b>Sorting</b> <ol style="list-style-type: none"> <li>a. Introduction</li> <li>b. Bubble Sort</li> <li>c. Insertion</li> <li>d. Selection</li> <li>e. Quick</li> <li>f. Merge</li> <li>g. Comparison and Efficiency of sorting</li> </ol>	2 2 1	5	7
8	<b>Searching</b> <ol style="list-style-type: none"> <li>a. Introduction</li> <li>b. Sequential Searching</li> <li>c. Binary Search</li> <li>d. Comparison and Efficiency of Searching</li> <li>e. Hashing           <ul style="list-style-type: none"> <li>• Probing (Linear and Quadratic)</li> </ul> </li> </ol>	2 2 1	5	7
9	<b>Graph</b> <ol style="list-style-type: none"> <li>a. Introduction</li> <li>b. Representation of Graph           <ul style="list-style-type: none"> <li>• array</li> <li>• linked list</li> </ul> </li> <li>c. Traversal           <ul style="list-style-type: none"> <li>• Depth First Search</li> <li>• Breadth First Search</li> </ul> </li> <li>d. Minimum spanning Tree           <ul style="list-style-type: none"> <li>• Kruskal's algorithm</li> </ul> </li> </ol>	1 2 1	4	5

**Text Book:**

Data structure using C and C++, Y. Langsam, M.J.Augenstein, A.M. Tenenbaum

**Reference Book:**

The Design and Analysis of Algorithm, Nitin Upadhyay, SK Kataria and Sons

**Prerequisite: C**