

## Semester II

<b>Year</b>	I	<b>Course Code:</b> 21BCA2C4L	<b>Credits</b>	03
<b>Sem.</b>	II		<b>Course Title:</b> Data Structures using C	<b>Hours</b>
Course Pre-requisites, if any	Knowledge of Programming			
Formative Assessment Marks: 40	Summative Assessment Marks: 60		Duration of ESA: 02 hrs.	
<b>Course Outcomes</b>	At the end of the course the student should be able to:  1. Understand the classification of data structures and dynamic memory allocation  2. Understand the difference between iteration and recursion and apply recursive definition for problem solving  3. Understand and evaluate the applications of stacks and queues  4. Understand and evaluate the applications of linked lists and tree			
<b>Unit No.</b>	<b>Course Content</b>		<b>Hours</b>	
Unit I	<b>Introduction to data structures:</b> Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - malloc, calloc, realloc and free. <b>Pointers in C:</b> Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;		08	
Unit II	<b>Recursion:</b> Definition; Types of recursions; Recursion Technique Examples - GCD, Binomial coefficient nCr, Towers of Hanoi; Comparison between iterative and recursive functions. <b>Sorting and Searching:</b> Arrays as abstract data types, Representation of linear arrays in memory, Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching		10	

Unit III	<p><b>Stacks:</b> Basic Concepts – Definition and Representation of stacks; Operations on stacks – Push, Pop; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.</p> <p><b>Queues:</b> Basic Concepts – Definition and Representation of queues; Types of queues, - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;</p>	10
Unit IV	<p><b>Linked list:</b> Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Circular linked list Doubly Circular Linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion;</p> <p><b>Trees:</b> Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal &amp; non-terminal nodes, degree of a node, level, edge, path, depth;</p> <p><b>Binary tree:</b> Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; preorder, in order and post order traversal.</p>	12
<b>Recommended Learning Resources</b>		
Print Resources	<ol style="list-style-type: none"> <li>1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures</li> <li>2. Tanenbaum: Data structures using C (Pearson Education)</li> <li>3. Kamathane: Introduction to Data structures (Pearson Education)</li> <li>4. Y. Kanitkar: Data Structures Using C (BPB)</li> <li>5. Kottur: Data Structure Using C</li> <li>6. Padma Reddy: Data Structure Using C</li> </ol>	

Year	I	Course Code: 21BCA2C4P	Credits	02
Sem.	II	Course Title: Lab: Data Structures	Hours	40
Course Pre-requisites, if any:	Knowledge of Programming			
Formative Assessment Marks: 25	Summative Assessment Marks: 25		Duration of ESA: 02 hrs.	
		<p style="text-align: center;"><b><u>Part A:</u></b></p> <p>1. Program to find GCD using recursive function</p> <p>2. Program to generate binomial coefficient using recursive function.</p> <p>3. Program to generate n Fibonacci numbers using recursive function.</p> <p>4. Program to implement Towers of Hanoi using recursion.</p> <p>5. Program to implement dynamic array, find smallest and largest element of the array.</p> <p>6. Program to read the names of cities and arrange them alphabetically.</p> <p>7. Program to sort the given list using selection sort technique.</p> <p>8. Program to sort the given list using bubble sort technique.</p> <p>9. Program to sort the given list using insertion sort technique.</p>		
		<p style="text-align: center;"><b><u>Part B:</u></b></p> <p>1. Program to sort the given list using quick sort technique.</p> <p>2. Program to sort the given list using merge sort technique.</p> <p>3. Program to search an element using linear search technique.</p> <p>4. Program to search an element using binary search technique.</p> <p>5. Program to implement Stack.</p> <p>6. Program to convert an infix expression to postfix.</p> <p>7. Program to implement simple queue.</p> <p>8. Program to implement linear linked list.</p> <p>9. Program to display in-order traversal of a binary tree.</p>		