PARVATHANENI BRAHMAYYA
SIDDHARTHA COLLEGE OF ARTS & SCIENCE
Autonomous
Siddhartha Nagar, Vijayawada–520010
Re-accredited at 'A+' by the NAAC

Course Code				**23CGMAL231**			
Title of the Course				**Data Structures**			
Offered to: (Programme/s)				**B. Sc. Hons (CSCS)**			
L	**4**	**T**	**0**	**P**	**0**	**C**	**3**
Year of Introduction:		**2024-25**		**Semester:**		**3**	
Course Category:		**Major Theory**		**Course Relates to:**		**Global / National / Regional / Local**	
Year of Revision:		**-**		**Percentage:**		**-**	
Type of the Course:				**Skill Development**			
Crosscutting Issues of the Course :							
Pre-requisites, if any				Basic knowledge of programming concepts Familiarity with the C programming language is recommended.			

Course Description: To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Understand various Data Structures for data storage and processing.
2	Realize Linked List Data Structure for various operations
3	Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures.
4	Understand and implement various searching & sorting techniques
5	Understand the Non-Linear Data Structures such as Binary Trees and Graphs

Course Outcomes

At the end of the course, the student will be able to…

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Describe and differentiate between various data structures and their uses.	K2	1, 2	1
CO2	Implement and manipulate data structures using C.	K3	1, 2	1

CO3	Analyze and evaluate the efficiency of algorithms.	**K4**	**7**	**1**
CO4	Solve complex problems by selecting and applying appropriate data structures.	**K3**	**7**	**1**
CO5	Demonstrate proficiency in dynamic memory management and pointer manipulation in C.	**K3**	**1, 2**	**1**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**3**	**2**						**2**	
CO2	**3**	**2**						**3**	
CO3							**3**	**3**	
CO4							**3**	**2**	
CO5	**2**	**2**						**2**	

Use the codes 3, 2, 1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure:

Data structures

Unit -1: Introduction to data structures: Types of data structures-Primitive data structures, Nonprimitive data structures – linear data structures, nonlinear data structures, real world applications of data structures, Abstract data types-ADT for stack, queue, linked list, Performance analysis of algorithms-time complexity, space complexity. (10Hrs)

Description:

Data structures are fundamental concepts in computer science and programming, designed to organize, manage, and store data efficiently. Understanding data structures is essential for solving complex problems and optimizing the performance of software.

Examples:

Time Complexity: Looking up a specific page number in a well-organized notebook. If you know the page number, you can go directly to that page without flipping through the rest of the notebook. The time taken is the same regardless of how many pages are in the notebook.

Space Complexity: Exchanging two items between your hands. No matter how large the items or how many times you swap, you only need a fixed amount of space (your two hands). Similarly, the algorithm only requires a constant amount of extra space, regardless of the input size.

Exercises

Program to insert, update, delete an element

Learning Outcomes:

Understand various Data Structures for data storage and processing.

Specific Resources: (web)

https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Unit – 2 : Linked List: (14Hrs)

Linked List: Introduction to Linked Lists, linked lists ADT, Comparison between Linked List and Array, Types of Linked Lists and their implementations - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Application of linked lists

Description:

Linear data structures are data structures where elements are arranged sequentially, one after another. In a linear data structure, each element has a unique predecessor and successor (except the first and last elements). These structures are simple and easy to implement, making them foundational in computer science.

Examples:

The university's administration requires a system to manage student records, which include operations such as adding, searching, updating, and deleting student records as well as deleting student reports

Exercises:

Implement Single Linked List with insertion, deletion and traversal operations

Learning Outcomes:

Realize Linked List Data Structure for various operations

Specific Resources: (web)

https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Unit – 3 : Stacks And Queues (14Hrs)

Introduction to stack, Stack ADT, stacks using array and Linked List, Application of stacks – Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation - Tower of Hanoi, Recursion

Introduction to Queue: Queue ADT, Queues using arrays and Linked List, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue

Description:

A stack is a linear data structure that follows the Last In, First Out (LIFO) principle. This means that the last element added to the stack will be the first one to be removed. Stacks are used in various applications, including algorithm implementation, memory management, and backtracking problems.

Examples:

To store data of books in a last-in, first-out (LIFO) manner.

An online bookstore needs to manage its inventory, process customer orders, and recommend books to users. To achieve these tasks efficiently, the bookstore must use various data structures.

Exercises:

Programs to implement the Queue operations using an array and linked Lists

Learning Outcomes:

Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures

Specific Resources: (web)

https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Unit – 4 : Searching and Sorting (8 Hrs)

Linear or Sequential Search, Binary Search and Indexed Sequential Search

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

Description:

Searching is the process of finding a particular element or a set of elements in a collection of data. It is a fundamental operation in computer science, crucial for various applications like databases, information retrieval, and algorithms

Examples:To search books based on user requirement such as specific book title , author etc …. Imagine an online bookstore where books are stored in an array or a list. If a user searches for a book by its title, the system can use linear search to find the book.

Exercises:

- program to search an item in a given list using Linear Search & Binary Search.
- Searching Algorithms
- program for implementation of Bubble Sort Insertion Sort Quick Sort Sorting Algorithms

Learning Outcomes:

Understand and implement various searching & sorting techniques.

Specific Resources: (web)
https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Unit – 5 : Trees and Graphs: (14Hrs)

Introduction to Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree. **Graphs:** Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs (DFS, BFS), Application of Graphs.

Description:

A binary tree is a hierarchical data structure in which each node has at most two children, referred to as the left child and the right child. Binary trees are used in various applications, such as searching, sorting, and representing hierarchical data like file systems.

Examples:

To search books based on user requirement such as ISBN or ISSN number. Imagine an online bookstore where books are stored in an array or a list. If a user searches for a book by its ISSN or ISBN number, the system can use binary search tree to store and retrieve the book based on unique keys.

Exercises:

program for Binary Search Tree Traversals

Learning Outcomes:

Understand the Non-Linear Data Structures such as Binary Trees and Graphs

Specific Resources: (web)
https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Text Books:

1. Horowitz and Sahani, “Fundamentals of Data Structures”, Galgotia Publications Pvt Ltd Delhi India.
2. A.K. Sharma ,Data Structure Using C, Pearson Education India.
3. “Data Structures Using C” Balagurusamy E. TMH

Reference Books

1. “Data Structures through C”, Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, “Data Structure Using C and C++” Wiley Dreamtech Publication. 3. Lipschutz, “Data Structures” Schaum’s Outline Series, Tata Mcgraw-hill Education (India)Pvt. Ltd .
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount “Data Structures and Algorithms in C++”, Wiley India.

@@@@

PARVATHANENI BRAHMAYYA
SIDDHARTHA COLLEGE OF ARTS & SCIENCE
Autonomous
Siddhartha Nagar, Vijayawada–520010
Re-accredited at 'A+' by the NAAC

SEMESTER -END QUESTION PAPER STRUCTURE

Course Code & Title of the Course:	**23CGMAL231 : DATASTRUCTURES**
Offered to:	**B. Sc. Hons (CSCS)**
Category:	**SEMESTER: 3**
Max. Marks	**70**
Max.Time	**3 Hrs**

Section A: Short Answer Questions (20 Marks)

Answer All questions. Each question carries 4 Marks.

1 a) Write ADT for stacks and explain it. K1
(or)
b) Write real world applications of data structures K1

2 a) Compare linked lists with arrays K2
(or)
b) Explain about different types of linked lists K2

3 a) Write differences between stacks and queues K2
(or)
b) Convert following expression from infix to postfix. K2
a+b*c+(d*e+f) +g.

4 a) Write program for linear search. K1
(or)
b) Write program for Bubble sort. K1

5 a) Discuss applications of graphs. K2
(or)
b) Explain with examples sequential and linked representation of graphs. K2

Long Answer Questions (50 Marks)

<u>Section B:</u>

Answer All questions. Each question carries 10 Marks.

6 a) Give classifications of Data structures and explain them. K2
(or)
b) Explain about analysis of algorithms. K2

7 a) Develop code for insertion and deletion in single linked list. K2
(or)
b) Write functions for insertion, display of values in doubly linked list. K2

8 a) Write code to implement queues using arrays. K2

(or)

b) Write code to implement stacks using linked list. K2

9 a) Write program for binary search. K2

(or)

b) Apply quick sort for below given values and write code to implement quick sort. K2

11 2 9 13 57 25 17 1 90 3.

10 a) Explain with code deletion in binary search tree. K2

(or)

b) Explain Depth first search with an example. K2

****