****



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **L** | **T** | **No. of PSDA** | **P/S** | **SW/FW** | **TOTAL CREDIT UNITS** |
| 3 | - | 5 | 2 | 2 | 5 |

**Course Title: Data Structure Using C**

**Course Code: CSIT 124**

**Credit units: -04**

Course level: -UG

**Course Type: -PC-Core**

**Course Objectives: -** The aim of this course is to

* Impart in-depth knowledge of data structure and its implementation in computer programs.
* Make students understand the concepts of linear and nonlinear data structure.
* Illustrate asymptotic notations and their usage.

**Pre-requisites: -** ‘C’ language

**Course Learning Outcomes:**

After successful completion of this course, the student will be able to

* Apply advance C programming techniques such as pointers, dynamic memory allocation, structures to developing solutions for problems.
* Development of stack and queue data structures for solving real world problems.
* Describe and implement abstract data types such as linked list, stack, queue and tree by using ‘C ‘ for static and dynamic implementations.
* Analyze, and evaluate appropriate abstract data types and algorithms to solve problems.
* Describe and implement trees by using ‘C’ for static and dynamic implementations,
* Analyze and implement graph theory and its applications.

**Course Contents: -**

|  |  |
| --- | --- |
|  | **Weightage (%)** |
| **Module I: Introduction to Data Structures** | **15** |
| Definition, Types. Algorithm design, Complexity, Time-Space Tradeoffs. Use of pointers in data structures.  Array Definition and Analysis, Representation of Linear Arrays in Memory, Traversing of Linear Arrays, Insertion And Deletion, Single Dimensional Arrays, Two Dimensional Arrays, Multidimensional Arrays, Function Associated with Arrays, Character String in C, Character String Operations, Arrays as parameters, Implementing One Dimensional Array, Sparse matrix. |
| **Module II: Stacks and Queues** | **15** |
| Definition, Array representation of stacks, Operations Associated with Stacks- Push & Pop, Polish expressions, Conversion of infix to postfix, infix to prefix (and vice versa),Application of stacks recursion, polish expression and their compilation, conversion of infix expression to prefix and postfix expression, Tower of Hanoi problem.  **Queue:** Definition, Representation of Queues, Operations of queues- Insert, Delete, Priority Queues, Circular Queue, Deque. |
| **Module III : Programming with Linked Lists** | **20** |
| Introduction to Singly linked lists: Representation of linked lists in memory, Traversing, Searching, Insertion into, Deletion from linked list, Garbage collection and compaction, doubly linked list, operations on doubly linked list, circular linked list, operations on circular linked list, generalized list. Applications of Linked List-Polynomial representation using linked list and basic operation. Stack and queue implementation using linked list. |
| **Module IV :Trees** | **20** |
| **Trees:** Basic Terminology, Binary Trees and their representation, expression evaluation, Complete Binary trees, extended binary trees, Traversing binary trees, Searching, Insertion and Deletion in binary search trees, General trees, AVL trees, Threaded trees, B trees. |  |
| **Module V: Searching and Sorting Techniques** | **15** |
| Insertion Sort, Bubble sort, Selection sort, Quick sort, Merge sort, Heap sort, Partition exchange sort, Shell sort, Sorting on different keys, External sorting. Linear search, Binary search, Hashing:,Hash Functions, Collision Resolution Techniques. |  |
| **Module VI:** **Graph and Their Applications** | **15** |
| Introduction, Graph Theory Terminology, Sequential Representation of Graph (Adjacency and Path Matrix), Warshall Algorithms, Linked Representation of Graph, Different Operations on Graphs, Traversing A Graph(DFS, BFS)., Spanning Trees-Introduction .Representation of Spanning tree, Constructing A Spanning Tree(Prim’s Algorithm, Kruskal’s Algorithm). |  |

**Pedagogy for Course Delivery: -** Subject will be taught based on lectures and practical will be conducted in blended/flipped mode. Particular emphasis will be given on practical explaining use case scenario for various algorithms. Focus will be on student’s involvement while imparting the course contents.

**Self-Work/ Professional Skill Development activities**

|  |  |
| --- | --- |
| **Module 1** | **Introduction to Data Structures**   1. What is Data Structure? What are their types and subtype? Explain each of subtype with examples in details. 2. Define O notation of time complexity. 3. Write a program in C++ which accepts an integer array and its size as arguments/parameter and assign the elements into two-dimension array of integers in the following format.   If the array is 1 2 3 4 5 6  1 2 3 4 5 6  1 2 3 4 5 0  1 2 3 4 0 0  1 2 3 0 0 0  1 2 0 0 0 0  1 0 0 0 0 0   1. Each element of an array DATA [1….10][1….10] requires 8 bytes of storage. If base address of array DATA is 2000, determine the location of DATA[4][5], when the array is stored   Row-wise  Column-wise   1. Write a function in C++, which accepts an integer array and its size as arguments and swap the elements of every location with its following odd location.    * Example: If an array of nine elements initially contains the elements as 24,1,6,7,9,23,10 then the function should rearrange the array as 4,2,6,1,7,5,23,9,10. 2. Write a program to calculate the upper triangle of a square matrix. 3. **Differentiate file structure from storage structure.** |
| **Module 2** | **Stacks and Queues**   1. Evaluate the following postfix expression using a stack and show the contents of stack after execution of each operation: 120,45,20,+,25,15,-,+,\* 2. Discuss the various applications of stacks. Also write an algorithm to PUSH and POP an element into the Stack. 3. What is difference between an array and a stack housed n an array? Why stack is called LIFO data structure? Explain how push and pop operation are implemented on a stack. 4. Write a program to implement Stack using array, also show overflow and underflow in respective push and pop operations. 5. Explain Push, Pop and Peep operation with algorithm. 6. What is Tower of Hanoi problem in C? Write a program to implement it. 7. Write a program to implement the operation of queue-Insert, Delete and explain the Priority Queues. 8. Write a program for circular queues. |
| **Module 3** | **Programming with Linked Lists**   1. What is the limitation of sequential data structures? 2. Define sparse matrix. 3. Write a program to implement Linear Linked List, showing all the operations, like creation, display, insertion, deletion and searching. 4. Write a program to implement Stack, using Linked List. Implement Push, Pop and display operations. 5. Write a program to implement Queue, using Linked List. Implement Insertion, deletion and display operations. 6. Write a program to count the number of times an item is present in a linked list. 7. Write a program to increment the data part of every node present in a linked list by 10. Display the data both before incrimination and after. 8. Write a program to implement Doubly Linked List, showing all the operations, like creation, display, insertion, deletion and searching. 9. Write a program to create a Binary Search Tree and display its contents using recursive preorder, postorder and inorder traversal.   **Trees**   1. Define tree, degree of node and siblings. 2. List out type’s binary tree. 3. What is the difference between full binary tree & complete binary tree? 4. Trace the binary tree of preorder traversal: PFBHGSRYTWZ. 5. Create a binary tree using in-order and pre-order traversal  Inorder : DBHEAIFJCG   Preorder : ABDEHC FIJG   1. Using the following binary tree traverse it into inorder, preorder and postorder:      1. Write an algorithm to delete an element x from a binary search tree t. Discuss your method with an example. 2. Write a program to create a Binary Search Tree and display its contents using preorder, postorder and inorder traversal. |
| **Module 4** | **Searching and Sorting Techniques**   1. Write down complexity of all sort and in which situation those sorts should be used? 2. Which sorting techniques is an application of recursion? 3. Use the selection sort to put the number 3, 2, 4, 1, 5 into increasing order. Illustrate the output returned in each pass clearly. Also, write the pseudo algorithm to it. 4. Write a program to sort the given array using MergeSort. 5. Write a program of Graph traversal-Depth first search and Breadth first search. 6. Trace quick sort on the list L= {11, 34,67, 78, 78, 78, 99}.What are your observations? 7. Write the program for heap sort. 8. What is hashing and hash function in data structure explain in detail. 9. What is a collision resolution technique in data structure? |
| **Module 5** | **Graph and Their Applications**   1. Write a program of Graph traversal-Depth first search and Breadth first search. 2. Construct a Spanning Tree and calculate the minimum cost using Prim’s and Kruskal’s Algorithm. 3. Explain Warshall Algorithms and write a program to implement it. 4. What is Sequential Representation of Graph (Adjacency and Path Matrix). 5. Explain different Operations on Graphs. |

.

**List of Experiments:**

1. Write a program to search an element using Linear Search.
2. Write a program to search an element using Binary Search.
3. Write a program to sort the given array using Bubble Sort.
4. Write a program to sort the given array using Selection Sort.
5. Write a program to sort the given array using Insertion Sort.
6. Write a program to sort the given array using QuickSort.
7. Write a program to sort the given array using MergeSort.
8. Write a program to insert a new element in the given unsorted array at kth position.
9. Write a program to delete an element from given sorted array.
10. Write a program to merge to given sorted arrays.
11. Write a program to implement Stack using array, also show overflow and underflow in respective push and pop operations.
12. Write a program to implement Queue using array, which shows insertion and deletion operations.
13. Write a program to implement Circular Queue using array, which shows insertion and deletion operations.
14. Write a program to implement Linear Linked List, showing all the operations, like creation, display, insertion, deletion and searching.
15. Write a program to implement Stack, using Linked List. Implement Push, Pop and display operations.
16. Write a program to implement Queue, using Linked List. Implement Insertion, deletion and display operations.
17. Write a program to count the number of times an item is present in a linked list.
18. Write a program to increment the data part of every node present in a linked list by 10. Display the data both before incrimination and after.
19. Write a program to implement Doubly Linked List, showing all the operations, like creation, display, insertion, deletion and searching.
20. Write a program to create a Binary Search Tree and display its contents using recursive preorder, postorder and inorder traversal.
21. Write a program to implement deletion of a node in binary search tree.
22. Write a program to implement Binary tree and display the contents using non-recursive preorder, postorder and inorder traversal techniques.
23. Write a program to sort the given array using HeapSort.
24. Write a program of Graph traversal-Depth first search and Breadth first search.

25. Write a program to implement Prim’s algorithm

26. Write a program to implement Kruskal algorithm.

* **Assessment/ Examination Scheme:**

|  |  |  |
| --- | --- | --- |
| **Theory L/T (%)** | **Lab/Practical/Studio (%)** | **Total** |
| **80** | **20** | **100** |

* **Theory Assessment (L&T):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Continuous Assessment/Internal Assessment** | | | |  | **End Term Examination** |
| **Components (Drop down)** | **Self-Work/PSDA** | **Mid Term** | **Viva** | **Attendance** | **EE** |
| **Weightage (%)** | 20 | 10 | 05 | 5 | 60 |

* **Lab/ Practical/ Studio Assessment:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Continuous Assessment/Internal Assessment** | | | | | **End Term Examination** | |
| **Components (Drop down** | **Attendance** | | **Mid term** | **Lab Record** | **Continuous Assessment** | **Viva** | **Practical** | **Viva** |
| **Weightage (%)** | 5 | | 10 | 10 | 10 | 5 | 40 | 20 |

## Text:-

* Yashwant Kanetkar,”Data Structure using C”, BPB Publication, 5th Edition ,2011
* A.Tannenbaum,Y. Lanhgsam and A.J. Augenstein ,” Data Structures Using C And C++ “,Prentice Hall of India,2nd Edition,2009.
* Jean-Paul Tremblay, P.G Sorenson, “An Introduction to Data Structures with applications”, Mcgraw-Hill ,2nd Edition ,1984.

***References:-***

* Robert L Kruse, “Data Structure and Program Design in C”, Prentice Hall (1991).
* Noel Kalicharan ,“Data Structure in C” ,Ist Edition Create space publisher, 2008.
* Mark Allen Weiss,“Data Structure and algorithm Analysis in C”,2nd Edition AddisonWesley,1996.
* E. Balagurusamy, “Problem Solving through C language”, TMH publication, Fourth Edition, 2008.
* R.S Salaria ,“Data Structures & Algorithms using C”,Khanna Publication,4th Edition,2009
* E.Horowitz and S.Sahni,”Fundamentals of Data Structures in C “,2nd Edition, Universities Press,2008.