|  |  |  |  |
| --- | --- | --- | --- |
| **Semester** | **III** | | |
| **Course Title:** | **Data Structures** | | |
| **Course Code:** | **23IC3PCDSC** | **Total Contact Hours: 40 hours** | |
| **L-T-P:** | **3-0-1** | **Total Credits:** | **4** |

|  |  |  |
| --- | --- | --- |
| **Unit No.** | **Topics** | **Hours** |
| 1 | **Introduction To Data Structure:** Data Management concepts, Data types – primitive and non-primitive, Types of Data Structures- Linear & Non-Linear Data Structures. Structures and pointers  **Dynamic memory allocation**: allocating a block of memory: Malloc, allocating multiple blocks of memory: Calloc, Releasing the used space: Free Altering the size of memory: Realloc. | **8** |
| 2 | **Linear list:** Singly linked list implementation, insertion, deletion and searching operations on linear list, circularly linked lists- insertion, deletion and searching operations for circularly linked lists, doubly linked list implementation, insertion, deletion and searching operations, maintaining directory of names, Manipulation of polynomials (addition), representing sparse matrices. | **8** |
| 3 | **Stacks:** Operations, array representations of stacks, stack applications - infix to postfix conversion, postfix expression evaluation, and function call tracing, recursion.  **Queues:** Introduction, Basic concept, linear queue operations, circular queue, priority queues, double ended queues. Applications of Queues.  Stack and queue implementation using linked lists | **8** |
| 4 | **Trees:** Definitions, tree representation, properties of trees, Binary tree, Binary tree representation, binary tree properties, binary tree traversals, binary tree implementation, Binary Search Tree operations and its implementation, applications of trees. | **8** |
| 5 | **Balanced Trees:** AVL Trees, Splay trees, Red- Black Trees – Definitions, Rotation and other basic operations. | **8** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Prescribed Text Book** | | | | | | | | | |
| **Sl. No.** | **Book Title** | | **Authors** | | **Edition** | **Publisher** | | | **Year** |
| 1. | Fundamentals of Data Structures in C | | Horowitz, Sahni, Anderson Freed | | Second | Universities Press | | | 2008 |
| 2. | Data Structures using C | | Reema Thareja | | Second | Oxford University press | | | 2014 |
| **Reference Text Book** | | | | | | | | | |
| **Sl. No.** | **Book Title** | **Authors** | | **Edition** | | | **Publisher** | **Year** | |
| 1. | Data Structures using C | Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein | | Fifth | | | Pearson Education | 2007 | |
| 2 | Data Structures - A Pseudocode Approach with C | Richard F. Gilberg  Behrouz A. Forouzan | | First | | | Cengage Learning | 2005 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **E-Book** | | | | | | |
| **Sl. No.** | **Book Title** | **Authors** | **Edition** | **Publisher** | **Year** | **URL** |
| **1.** | Data Structures using C | E. Balaguruswamy |  | McGraw Hill | 2013 | <https://dokumen.pub/data-structures-using-c-9781259029547-1259029549.html> |
| **2.** | Data structures and  program design in C | Robert L. Kruse, Clovis L.  Tondo, Bruce P. Leung | Second | Prentice Hal | 1997 | <https://cdn.preterhuman.net/texts/math/Data_Structure_And_Algorithms/Data%20Structures%20and%20Program%20Design%20in%20C++%20-%20Robert%20L.%20Kruse.pdf> |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MOOC Courses** | | | | |
| **Sl. No.** | **Course name** | **Course Offered By** | **Year** | **URL** |
| **1** | Data Structures | Coursera | 2023 | https://www.coursera.org /learn/data-structures |
| **2** | Data Structures and Algorithms | NPTEL | 2023 | https://nptel.ac.in/ courses/106102064/ |

**Course Outcomes**

***At the end of the course the student will be able to***

|  |  |
| --- | --- |
| **CO1** | Apply the concept of linear and nonlinear data structures for computing problems. |
| **CO2** | Analyse the appropriate data structure operations for a given problem |
| **CO3** | Design and develop solutions using the linear and nonlinear data structure for a given specification. |
| **CO4** | Conduct experiments for demonstrating the operations of different data structures. |

**CO-PO mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** |  | 3 |  |  |  |  |  |  |  |  |  |  |
| **CO3** |  |  | 3 |  |  |  |  |  |  |  |  |  |
| **CO4** |  |  | 3 |  | 3 |  |  |  |  | 1 |  |  |

**Proposed Assessment Plan (for 50 marks of CIE)**

|  |  |  |
| --- | --- | --- |
| **Assessment Tool** | **No. of Assessments** | **Marks** |
| Internals | 2 | 20 |
| QUIZ/AAT | 1 | 5 |
| Lab Component | CIE+ Two Lab Tests | 25 |
| **Total** | | **50** |

**Laboratory Plan**

**Instructions to Students to be followed in each lab:**

1. Each Student should write down the program in the observation book and get it evaluated by the respective lab faculty in-charge and then execute the program.

2. Each Student should bring the lab record with the programs and output written for the programs completed in their respective previous week and get it evaluated by the lab faculty in-charge. In the record book students should - Handwrite the Program - Pasting of the printout of the Output or Handwriting of the Output (Output should be written for all the cases).

3. Students have to practice following list of programs and additional programming exercises will also be given in lab. Students will be made to solve coding challenges on programming platforms like LeetCode and HackerRank.

|  |  |  |
| --- | --- | --- |
| **Lab Program** | **Unit#** | **Program Details** |
| 1 | 2 | Write a program to implement Singly Linked List with following operations  a) Create a linked list.  b) Insertion of a node at first position, at any position and at end of list.  c) Display the contents of the linked list. |
| 2 | 2 | Write a program to Implement Singly Linked List with following operations  a) Create a linked list.  b) Deletion of first element, specified element and last element in the list.  c) Display the contents of the linked list. |
| 3 | 2 | Write a program to Implement Singly Linked List with following operations  a) Sort the linked list.  b) Reverse the linked list.  c) Concatenation of two linked lists |
| 4 | 2 | Write a program to Implement doubly linked list with primitive operations  a) Create a doubly linked list.  b) Insert a new node to the left of the node.  c) Delete the node based on a specific value  d) Display the contents of the list |
| 5 | 3 | Write a program to simulate the working of stack using an array with the following: a) Push b) Pop c) Display  The program should print appropriate messages for stack overflow, stack underflow |
| 6 | 3 | Write a program to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), \* (multiply) and / (divide) |
| 7 | 3 | Write a program to simulate the working of a queue of integers using an array. Provide the following operations  a) Insert b) Delete c) Display  The program should print appropriate messages for queue empty and queue overflow conditions |
| 8 | 3 | Write a program to simulate the working of a circular queue of integers using an array. Provide the following operations. a) Insert b) Delete c) Display  The program should print appropriate messages for queue empty and queue overflow conditions |
| 9 | 3 | Write a program to implement Stack & Queues using Linked Representation |
| 10 | 4 | Write a program  a) To construct a binary Search tree.  b) To traverse the tree using all the methods i.e., in-order, preorder and post order  c) To display the elements in the tree. |
| 11 | 4 | Write a program   1. To construct a binary search tree 2. To implement iterative inorder traversal 3. To delete a given element |
| 12 | 5 | Write a program to construct an AVL tree of integers |

**SEE Question paper format**

|  |  |  |
| --- | --- | --- |
| **Unit-1** | Mandatory | One Question to be asked for 20 Marks |
| **Unit-2** | Internal Choice | Two Questions to be asked for 20 Marks each |
| **Unit-3** | Mandatory | One Question to be asked for 20 Marks |
| **Unit-4** | Internal Choice | Two Questions to be asked for 20 Marks each |
| **Unit-5** | Mandatory | One Question to be asked for 20 Marks |