JGI JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

**PROGRAMME: MCA (GENERAL)**
**SEMESTER – I**
**Teaching-Learning & Evaluation Plan (TLEP)**

**Course Information:**

| | | |
|---|---|---|
| Course Code: **23MCAC101** | Course Title: Data Structures | |
| Credits Units: 04 | Contact Hours: 45 | L-T-P-E: 3-1-0-4 |
| IA: UE Weightage – 50 : 50 | Pass Marks(CA & UE)-40% and 40% | Aggregate Pass Marks: 40% |
| UE Question Paper Marks: 100 | Special Examination Fees: NA | |

**Prerequisite (if any):** Problem solving Techniques.
**Course Facilitator (s):** **Ms. Ninitha Mary Antonina N.M**, Assistant Professor, School of CS & Engineering.

**Programme Outcomes (POs) and Programme Specific Outcomes (PSOs)**

| At the end of the Programme, students will be able to | |
|---|---|
| **PO 1** | **Computational Knowledge:** Apply mathematical foundation, computing and domain knowledge for the conceptualization of computing models from defined problems. |
| **PO 2** | **Problem Analysis:** Analyze complex computing problems using fundamentals of computer science and application domains. |
| **PO 3** | **Design & Develop Solutions:** Transform complex business scenarios and contemporary issues into problems, and propose integrated solutions using emerging technologies. |
| **PO 4** | **Conduct Investigations of Complex Computing Problems:** Conduct experiments, interpret data and provide well informed conclusions. |
| **PO 5** | **Modern Tool Usage:** Select modern computing tools, skills and techniques necessary for providing innovative software solutions. |
| **PO 6** | **Innovation and Entrepreneurship:** Create ideas for opportunities, entrepreneurship vision and use of wealth for the betterment of the individual and society. |
| **PO 7** | **Societal & Environmental Concern:** Recognize economical, environmental, social, health, legal, ethical issues involved in the usage of computing technologies and other consequential responsibilities relevant to professional practice. |
| **PO 8** | **Professional Ethics:** Employ professional ethics and cyber regulations in a global economic environment. |
| **PO 9** | **Individual & Team Work:** Function effectively as a member or leader in diverse teams in a multidisciplinary environment. |
| **PO 10** | **Communication Efficacy:** Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations. |
| **PO11** | **Project Management:** Demonstrate managerial skills with computing knowledge to manage projects in multidisciplinary environments. |
| **PO12** | **Life-long Learning:** Develop the ability and acquire the skills required to engage in continuous learning as a Computing professional. |
| **PSO1** | Acquire proficiency in developing and implementing effective solutions using new age IT platforms and emerging technologies. |
| **PSO2** | Develop professional and ethical attitude, effective communication skills, team work skills |

| | |
|---|---|
| | and multidisciplinary approach to relate computer solutions to broader social context thereby enhancing entrepreneurship skills. |

**Course Objectives:**

| | |
|---|---|
| **COB1** | To introduce the different types of Data Structures. |
| **COB2** | To help students by providing effective data access by organizing data effectively. |

**Course Outcomes:**

At the end of the course, students will be able to

| Sl. No. | Course Outcome | Description | Bloom's Taxonomy Level | PO's | Assessment |
|---|---|---|---|---|---|
| 1 | **CO 1** | Select Linked Lists or Arrays based on the time-space complexity required in the expected solution. | **L2** | PO1, PO2, PO3,PSO1 | IA-1 Activity-1 |
| 2 | **CO 2** | Apply the Stacks and Queues sequential data structures in solving real world problems | **L 3** | PO1, PO2, PO3, PO4, PO5, PO8, PO11, PO12, PSO1 | IA – 1 Activity - 1 |
| 3 | **CO 3** | Experiment with General and Binary Trees on accessing the data in random fashion using traversal techniques. | **L 4** | PO1, PO2, PO3, PO4, PO6, PO8, PO9, PO11, PO12, PSO1 | IA – 1 Activity - 1 |
| 4 | **CO 4** | Evaluate the performance of different types of Trees, Hashing and Heaps on solving real time problems. | **L 5** | PO1, PO2, PO3, PO4 PO6, PO9, PO10, PO11, PO12, PSO1 | IA – 2 Activity - 2 |
| 5 | **CO 5** | Develop the solutions using Graphs and Hashing for real world problems. | **L 6** | PO1, PO2, PO3, PO4 PO6, PO9, PO10, PO11, PO12, PSO1 | IA – 2 Activity - 2 |

**CO-PO/PSO Mapping:** (3-Strong Correlation 2- Medium Correlation 1- Low Correlation)

| Course Outcome | Bloom's Taxonomy Level | Program Outcomes (PO) | Program Specific Outcomes (PSO) |
|---|---|---|---|

| | | PO 01 | PO 02 | PO 03 | PO 04 | PO 05 | PO 06 | PO 07 | PO 08 | PO 09 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | L3 | 2 | 3 | 1 | - | - | - | - | - | - | - | - | - | 2 | 1 |
| CO2 | L3 | 2 | 2 | 1 | 3 | 2 | - | - | 2 | | - | 1 | 2 | 2 | 1 |
| CO3 | L3 | 3 | 2 | 3 | 3 | - | 1 | - | 2 | 1 | - | 2 | 2 | 2 | 3 |
| CO4 | L4 | 3 | 1 | 3 | 1 | - | 1 | - | - | 1 | 1 | 2 | 3 | 3 | 3 |
| CO5 | L3 | 3 | 2 | 3 | 2 | - | 1 | - | - | 1 | 2 | 2 | 3 | 3 | 3 |
| | Total | 13 | 10 | 11 | 9 | 2 | 3 | - | 4 | 3 | 3 | 7 | 10 | 12 | 11 |
| | | 2.6 | 2 | 2.2 | 2.3 | 2 | 1 | - | 2 | 1 | 1.5 | 1.8 | 2.5 | 2.4 | 2.3 |

**Course Contents:**

| Module | Details | Contact Hours |
|---|---|---|
| I | **List ADT**<br>Abstract Data Types, List ADT, Static and Dynamic Arrays, Array Operation. Linked List Implementation, Singly Linked List, Circularly Linked List, Doubly Linked List- all operations–Creation, Insertion, Deletion, Search and Traversal, Circular Linked Lists, Applications of Linked Lists-Polynomial addition and subtraction – Sparse Matrix Implementation. | 12 |
| II | **Stacks & Queues**<br>Stack ADT- Stack operations Push, Pop, Peek, isEmpty, isFull and initialize, Stack implementation using Array, Stack implementation using Linked List, applications of Stacks, evaluating arithmetic expressions, conversion of infix to postfix expression.<br>Queue ADT – Enqueue, Dequeue, Peek, isEmpty, isFull and initialize, Queue implementation using Array, Queue implementation using Linked List, applications of Queue, Priority Queue — Circular Queue and Queue Applications. | 12 |
| III | **Trees**<br>Introduction to Tree Data Structure and Basic Terminology. Implementation of Tree ADT. Types of Trees. Binary Tree ADT, Enumeration of Binary Trees, Tree Traversal – Expression Trees – applications of trees – Binary Search Tree ADT. Construct BST from given preorder traversal, Binary<br>Tree to Binary Search Tree Conversion. | 12 |
| IV | **Advanced Trees and Hashing**<br>Binary Search Tree- Operations. AVL trees - Threaded Binary Tree, B Trees, B+ Tree, Heaps- Types of Heaps, Operations and Applications of heap.<br>Hashing – Hashing functions – Collision Strategy. | 12 |

| | | |
|---|---|---|
| V | **Graphs Data Structure**<br>Introduction to graph representation and Terminology – Types of Graphs— Graph traversal using Stack and Queue- Applications of Depth First and Breadth First Traversal, applications of graph, Detect Cycle in a Directed Graph and in an undirected graph, Transitive Closure of a Graph using DFS. | 12 |
| **Text Books (TB):** | | |
| TB | 1. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education Asia, 4th Edition, 2013. ISBN: 9780132871174 (Module: 1 to 5) | |
| **Reference Books (RB):** | | |
| RB | 1. Advanced Data Structures, Peter Brass, Cambridge University Press; 1st edition, 2008.<br>2. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, , University Press India Private Limited, Second Edition, 2005 | |
| **Other reading material:(Lab sessions)** | | |
| ORB | 1. The C Programming Language" by Brian Kernighan and Dennis Ritchie<br>2. "C Programming: A Modern Approach" by K.N. King<br>3. "Data Structures and Algorithmic Thinking with C++" by Narasimha Karumanchi | |

**Session-Wise Plan:**

| Abbreviations & Expansions | | | |
|---|---|---|---|
| **Pedagogy/Activity planned** | | **Mode of Delivery** | |
| **P1** | PPT presentation & Classroom interaction | **M1** | Synchronous – PPT |
| **P2** | Blended Learning with Hands on | **M2** | Asynchronous / synchronous |
| **P3** | Flip Class & Module Quiz | **M3** | Synchronous - Hands on |
| **A1** | **Activity-1** | **M4** | Synchronous - Discussion |
| **A2** | **Activity-2** | | |

| Module | Session No | Topic Name | No. of sessions | Readings and References | Pedagogy/ Activity planned | CO Mapping | Mode of Delivery |
|---|---|---|---|---|---|---|---|
| 1 | 1 | **List ADT**<br>Abstract Data Types, List ADT, Static and Dynamic Arrays | 1 | TB1<br>RB1<br>ORB1<br>WVL-1<br>WTL-1 | Students have to do a minimum 5 hours of any MOOC certification. | CO1 | Synchronous, PPT |
| | 2 | Array Operation | 1 | | | | |
| | 3 | Linked List Implementation: Singly Linked List:all operations– Creation, Insertion, Deletion, Search and Traversal | 1 | | | | |
| | 4 | Linked List Implementation: Singly Linked List:all operations– Creation, Insertion, Deletion, Search and Traversal | 1 | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **5** | Circularly Linked List : all operations–Creation, Insertion, Deletion, Search and Traversal | 1 | | | | |
| | **6** | Circularly Linked List : all operations–Creation, Insertion, Deletion, Search and Traversal | 1 | | | | |
| | **7** | Doubly Linked List: all operations–Creation, Insertion, Deletion, Search and Traversal | 1 | | Relate appropriate data structure to solve the given computational problem. | **CO1** | **Asynchronous, Blended learning with Hands on** |
| | **8** | Doubly Linked List: all operations–Creation, Insertion, Deletion, Search and Traversal | 1 | | | | |
| | **9** | Applications of Linked Lists-Polynomial addition and subtraction | 1 | | | **CO1** | **Experiential Lab sessions** |
| | **10** | Sparse Matrix Implementation. | 1 | | | | |
| | **11** | Applications of Linked Lists-Polynomial addition and subtraction | 1 | | | | |
| | **12** | Sparse Matrix Implementation. | 1 | | | | |
| | | | | | | | |
| **2** | **1** | **Stacks & Queues** Stack ADT | 1 | TB1<br>RB1<br>WVL-2<br>WTL-2 | Students have to do a minimum 5 hours of any MOOC certification. | **CO2** | **Synchronous, PPT Experiential Lab sessions Asynchronous, Blended learning with Hands on** |
| | **2** | Stack operations Push, Pop, Peek, isEmpty, isFull and initialize | 1 | | | | |
| | **3** | Stack implementation using Array | 1 | | | | |
| | **4** | Stack implementation using Linked List | 1 | | | | |
| | **5** | Applications of Stacks | 1 | | | | |
| | **6** | Evaluating arithmetic expressions | 1 | | | | |
| | **7** | Conversion of infix to postfix expression | 1 | | | | |
| | **8** | Queue ADT – Enqueue, Dequeue, Peek, isEmpty, isFull and initialize | 1 | | | | |
| | **9** | Queue implementation using Array | 1 | | | | |
| | **10** | Queue implementation using Linked List | 1 | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **11** | Applications of Queue, Priority Queue | 1 | | | | |
| | **12** | Circular Queue and Queue Applications. | 1 | | | | |
| | | | | | | | |
| **3** | **1** | **Trees** Introduction to Tree Data Structure. | 1 | TB1 RB1 WVL-3 WTL-3 | **Mini Project** | **CO3** | **Synchronous, PPT Experiential Lab sessions Asynchronous, Blended learning with Hands on** |
| | **2** | Basic Terminology | 1 | | | | |
| | **3** | Implementation of Tree ADT | 1 | | | | |
| | **4** | Types of Trees. Binary Tree ADT | 1 | | | | |
| | **5** | Enumeration of Binary Trees | 1 | | | | |
| | **6** | Tree Traversal | 1 | | | | |
| | **7** | Expression Trees – applications of trees | 1 | | | | |
| | **8** | Binary Search Tree ADT | 1 | | | | |
| | **9** | Construct BST from given preorder traversal | 1 | | | | |
| | **10** | Construct BST from given preorder traversal:Program discussion | 1 | | | | |
| | **11** | Binary Tree to Binary Search Tree Conversion. | 1 | | | | |
| | **12** | Binary Tree to Binary Search Tree Conversion: Program discussion | 1 | | | | |
| | | | | | | | |
| **4** | **1** | **Advanced Trees and Hashing** Binary Search Tree- Operations. | 1 | TB1 RB1 WVL-4 WTL-4 | **Mini Project** | **CO4, CO5** | **Synchronous, PPT Experiential Lab sessions Asynchronous, Blended learning with Hands on** |
| | **2** | AVL trees | 1 | | | | |
| | **3** | AVL trees | 1 | | | | |
| | **4** | Threaded Binary Tree | 1 | | | | |
| | **5** | B Trees | 1 | | | | |
| | **6** | B Trees | 1 | | | | |
| | **7** | B+ Tree | 1 | | | | |
| | **8** | Heaps- Types of Heaps | 1 | | | | |
| | **9** | Operations and Applications of heap | 1 | | | | |
| | **10** | Hashing | 1 | | | | |
| | **11** | Hashing functions | 1 | | | | |
| | **12** | Collision Strategy | 1 | | | | |
| | | | | | | | |
| **5** | **1** | **Graphs Data Structure** Introduction to graph | 1 | TB1 RB1 | **Mini Project** | **CO5** | **Synchronous, PPT Experi** |
| | **2** | Representation and Terminology | 1 | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **3** | Types of Graphs | 1 | WVL-5<br><br>WTL-5 | | | **ential Lab session s Async hronou s, Blende d learnin g with Hands on** |
| | **4** | Types of Graphs | 1 | | | | |
| | **5** | Graph traversal using Stack | 1 | | | | |
| | **6** | Graph traversal using Queue | 1 | | | | |
| | **7** | Depth First Traversal | 1 | | | | |
| | **8** | Applications of BFS and DFS | 1 | | | | |
| | **9** | Breadth First Traversal | 1 | | | | |
| | **10** | Applications of graph | 1 | | | | |
| | **11** | Detect Cycle in a Directed Graph and in an undirected graph | 1 | | | | |
| | **12** | Transitive Closure of a Graph using DFS. | 1 | | | | |
| | | | | | | | |

**Web Video Links:**

| | | |
|---|---|---|
| WVL-1 | Module-1 | https://www.youtube.com/watch?v=3Hsf2LwjjQE<br>https://www.youtube.com/watch?v=Tzw2rR_KeFg |
| WVL-2 | Module-2 | https://www.youtube.com/watch?v=-h5KziRac8I<br>https://www.youtube.com/watch?v=okr-XE8yTO8 |
| WVL-3 | Module-3 | https://www.youtube.com/watch?v=pYT9F8_LFTM<br>https://www.youtube.com/watch?v=-DzowlcaUmE<br>https://www.youtube.com/watch?v=1-l_UOFi1Xw |
| WVL-4 | Module-4 | https://www.youtube.com/watch?v=KyUTuwz_b7Q<br>https://www.youtube.com/watch?v=G46P8cHyDlo<br>https://www.youtube.com/watch?v=CVA85JuJEn0 |
| WVL-5 | Module-5 | https://www.youtube.com/watch?v=59fUtYYz7ZU<br>https://www.youtube.com/watch?v=j0IYCyBdzfA |

**Web Text Links: WTL**

| | | |
|---|---|---|
| WTL-1 | Module-1 | 1. https://www.javatpoint.com/ds-types-of-linked-list<br>2. https://www.w3schools.com/dsa/dsa_data_linkedlists_types.php |
| WTL-2 | Module-2 | 1. https://www.geeksforgeeks.org/introduction-to-stack-data-structure-and-algorithm-tutorials/<br>2. https://www.javatpoint.com/array-representation-of-queue |
| WTL-3 | Module-3 | 1. https://www.geeksforgeeks.org/binary-search-tree-data-structure/<br>2. https://www.javatpoint.com/binary-tree-vs-binary-search-tree |
| WTL-4 | Module-4 | 1. https://www.javatpoint.com/hashing-in-data-structure<br>2. https://www.javatpoint.com/tree<br>3. https://www.w3schools.com/dsa/dsa_theory_trees.php |
| WTL-5 | Module-5 | 1. https://www.geeksforgeeks.org/introduction-to-graphs-data-structure-and-algorithm-tutorials/<br>2. https://www.w3schools.com/dsa/dsa_theory_graphs.php<br>3. https://www.programiz.com/dsa/graph |

**Blended Learning [P2-Blended Learning with Hands on] :**
**Interaction - (30 minutes) Teacher Input 20 minutes) Wrap Up (10 minutes)**

| Sl. No | LinkedIn Learning Link |
|---|---|
| 1 | https://www.linkedin.com/learning/nail-your-c-plus-plus-interview/linked-lists-in-c-plus-plus?u=92695330 |
| 2 | https://www.linkedin.com/learning/nail-your-c-plus-plus-interview/stacks-in-c-plus-plus?u=92695330 https://www.linkedin.com/learning/nail-your-c-plus-plus-interview/queues-in-c-plus-plus?resume=false&u=92695330 |
| 3 | https://www.coursera.org/learn/data-structures?specialization=data-structures-algorithms |
| 4 | https://www.coursera.org/learn/data-structures?specialization=data-structures-algorithms |
| 5 | https://www.coursera.org/learn/data-structures?specialization=data-structures-algorithms |

**Case studies**

a. Real World case studies (https://www.programiz.com/blog/dsa-in-everyday-life/)

**A1 - Activity-1: MOOC certification**

Students have to do a minimum 5 hours of any MOOC certification.
Outcome: Students should be familiar with the concepts of data structures.

**Rubrics: A1**

| Criteria | **Poor** Below Expectation | **Fair** Needs improvement | **Good** Meets Expectations | **Excellent** Exceeds Exceptions | **Weightage** |
|---|---|---|---|---|---|
| Report Submission | *Poor:* Trivial Idea, No Clarity in items presented ***(5 Marks)*** | *Moderate:* Trivial Idea, No much Clarity in items presented ***(8 Marks)*** | *Good Report:* Trivial Idea, Clarity in items presented ***(12 Marks)*** | *Very Good Report:* Innovative Idea, Clarity in items presented ***(15 Marks)*** | 15 |

**A2 - Activity-2: MiniProject-2**

Students have to create any one of the given projects based in groups based on their roll numbers.

- Plagiarism detection system
- Banking management system
- Travel planner using Graph
- Building a crossword puzzle game
- Email Spam Filter

**Rubrics: A2**

| Criteria | **Poor** Below Expectation | **Fair** Needs improvement | **Good** Meets Expectations | **Excellent** Exceeds Exceptions | **Weightage** |
|---|---|---|---|---|---|
| Project Execution | *Poor:* Errors & No database operations ***(4 Marks)*** | *Moderate:* Errors & data database operations not up to the mark ***(5-6 Marks)*** | *Good Execution:* No Errors but database operations not up to the mark ***(7-8 Marks)*** | *Very Good Execution:* No Errors & good database operations ***(9-10 Marks)*** | 10 |
| Report Submission | *Poor:* Trivial Idea, No Clarity in items presented ***(2Marks)*** | *Moderate:* Trivial Idea, No much Clarity in items presented ***(3 Marks)*** | *Good Report:* Trivial Idea, Clarity in items presented ***(4 Marks)*** | *Very Good Report:* Innovative Idea, Clarity in items presented ***(5 Marks)*** | 5 |

**Assessment Scheme: IA: UE - 50:50**

| **Sl. No.** | **Assessment Instrument** | **Formative/ Summative** | **Frequency** | **Weightage (%)** | **CO** |
|---|---|---|---|---|---|
| 1 | MOOC Course | Formative | 1 | 15 | CO 1, CO 2, |
| 2 | Mini Project | Formative | 1 | 15 | CO 3, CO 4, CO 5 |
| 3 | Internal Test | Formative | 2 | 15 | CO 1, CO 2, CO 3, CO 4, CO 5 |
| 4 | Class Participation | Formative | -- | 5 | |
| 5 | University Exam | Summative | 1 | 50 | CO 1, CO 2, CO 3, CO 4, CO 5 |
| | **Total** | | | 100 | |

## Lab sessions(LS)

Course Title: **Data Structures Lab**

Course Code: **23MCAC101L**

Credits: **1** Contact Hours: **30** L:T:P:E 0**:0:2:0**

CA: UE Weightage - 100 Pass Marks (CA& UE) – **40%** Aggregate Pass Marks: **50 %**

### COURSE OBJECTIVES:

| Course Objectives | |
|---|---|
| **COB1** | To write programs to demonstrate the Linear and Non-Linear Data structures operations |
| **COB2** | To understand how the nodes are visited in the Non-Linear Data structures |
| **COB3** | To apply suitable data structures in developing applications. |

### COURSE OUTCOMES:

**After the completion of the course the students are able to**

| Course Outcomes | Description | Bloom's Taxonomy Level |
|---|---|---|
| **CO1** | Compare different types of Linked Lists, Stacks and Queues on creating the sequential data structure operations and applications. | L4 |
| **CO2** | Order the nodes in the Binary Trees to demonstrate how tree traversals can be done. | L5 |
| **CO3** | Produce the visiting of Graph nodes by using Breadth-First and Depth-First traversal techniques. | L6 |

## List of Programs

1. **Write a C program that uses functions to perform the following:**

a. Create a singly linked list of integers.

b. Delete a given integer from the above linked list.

c. Display the contents of the above list after deletion.

2. **Write a C program that uses functions to perform the following:**

a. Create a doubly linked list of integers.

b. Delete a given integer from the above doubly linked list.

c. Display the contents of the above list after deletion.

3. **Write C programs to implement a Stack ADT using i) Array and ii) Linked list respectively.**

4. **Write a C program that uses stack operations to convert a given infix expression into its**

**postfix equivalent.**

5. **Write C programs to implement a queue ADT using i) Array and ii) Linked list respectively.**
6. **Write a C program to implement a circular queue along with different operations using linked lists.**
7. **Write a C program that uses functions to perform the following:**

a. Create a Binary tree of numbers

b. Define functions to perform Inorder, Postorder and Preorder traversals on the above tree.

8. **Write a C program that uses functions to perform the following:**

a. Create a Binary Search Tree (BST) of integers.

b. Define a function to search for a given key in the BST.

9. **Write a C program to demonstrate an AVL Tree (Insertion and Rotations).**
10. **Write C programs for implementing the Breadth first graph traversal technique.**
11. **Write C programs for implementing the Depth first graph traversal technique.**
12. **Write C programs for demonstrating Hashing technique with Searching operation**