# Aror University of Art, Architecture, Design & Heritage Sukkur.

## BS(Artificial Intelligence)

**Fall-2024**

Data Structures (Theory + Lab)

Course Title: Data Structures (Theory + Lab)
Course Code: CSC-221
Credit Hours: (3+1)
Course Instructor: Abdul Ghafoor
Electronic mail: aghafoor.faculty@aror.edu.pk

**Course Objectives**

The objective of this course is to make students familiar with the concepts of the way data is stored inside computer and its manipulation using different algorithms. Students will learn different data structures such as array, stack, queue, linked list, trees, graphs, sorting algorithm etc. Since Programming fundamentals is the pre-requisite of this course, therefore, in class we would be using java language to implement all the data structures. However students may use any programming language.

## Assessment:

| S. No | **Assessment Activities** | Percentage | Total Activities |
|---|---|---|---|
| 1. | Sessional: Quizzes/ Assignments (Quizzes & Assignments) | 30% | 10 |
| 2. | Mid Term Exam | 30% | 1 |
| 3. | Final Exam | 40% | 1 |

# Aror University of Art, Architecture, Design & Heritage Sukkur.

---

**Course content:**

| Week No | Topics | Chapters |
|---|---|---|
| 1,2 | • Introduction to the course<br>• What is data structure?<br>  ○ Need of data structures<br>• Elementary data structures<br>• Arrays<br>  ○ Review of single-dimension arrays<br>  ○ Concept and implementation of 2D arrays<br>  ▪ Manipulating matrices using arrays<br>  ○ Basic concepts of Multi-dimensional arrays<br>  ○ What are limitations of Arrays? | See the chapter 3.1 in Michael T. Goodrich, Data Structures & Algorithms in Java |
| 3 | • Linked lists<br>  ○ Arrays vs. Linked list<br>• Types of linked list<br>  ○ Singly linked list<br>  ○ Circular singly linked list<br>  ○ Doubly linked list<br>  ○ Circular doubly linked list<br>• Defining the Node class<br>• Linked Lists Functions<br>Printing linked list in reverse order using recursion | See the chapter 3.2, 3.3, 3.4 in Michael T. Goodrich, Data Structures & Algorithms in Java<br><br>See the chapter 10.2 in [CLRS] Thomas H. Cormen, Introduction to Algorithms |
| 4 | • Applying dictionary operations on linked lists<br>  ○ Traversing a linked list<br>  ○ Inserting new node<br>  ▪ at the head<br>  ▪ at any location<br>  ○ Searching a node<br>  ○ Removing a node<br>  ▪ from the head<br>  ▪ from anywhere<br>• Clearing a linked list | Handouts |
| 5 | • Introduction to Queues<br>• The Queue data structure<br>• Application of queues<br>• Array Representation of Queue<br>  ○ Algorithm for Addition of an Element to the Queue<br>  ○ Algorithm for Deletion of an Element to the Queue<br>• Dynamic Representation of Queues Using Linked Lists | See the chapter 6.2 in Michael T. Goodrich, Data Structures & Algorithms in Java<br><br>See the chapter 10.1 in [CLRS] Thomas H. Cormen, Introduction to Algorithms |

| | | |
|---|---|---|
| | • Circular Queue-Array Representation | |
| 6 | • The FIFO structure<br>• Queue operations<br>• Extended queue operations<br>• Dictionary operations on queues<br>• The priority queues<br>• The LIFO structure<br>• Introduction to the stack data structure<br>• Applications of stack<br>• Stack operations<br>• Stack specifications<br>  ○ List and arrays<br>  ○ Stacks<br>  ○ Reversing a list<br>• Stack implementation<br>  ○ Using arrays<br>  ○ Using linked list<br>• Methods of stack<br>  ○ Push<br>  ○ Pop<br>• Push down stack | See the chapter 6.1 in Michael T. Goodrich, Data Structures & Algorithms in Java<br><br>See the chapter 10.1 in [CLRS] Thomas H. Cormen, Introduction to Algorithms |
| 7 | • What is algorithm?<br>• Complexity of algorithm<br>  ○ Time complexity<br>  ○ Space complexity<br>• Analysis of algorithms<br>• Big O Notation<br>  ○ Best-case analysis<br>  ○ Worst-case analysis<br>• Average-case analysis<br>• Recursion | Handouts |
| 8-9 | • Trees Introduction<br>• Tree terminology<br>• Tree Traversal<br>• Concept of Binary Trees<br>• Why use binary trees<br>• Basic Operations<br>• Complete Binary Tree<br>• Priority Queues: Heaps<br>• Max-Heap | See the chapter 8 in Michael T. Goodrich, Data Structures & Algorithms in Java<br><br>See the chapter 10.4 and 6 in [CLRS] Thomas H. Cormen, Introduction to Algorithms |
| 10-11 | • Concept of Binary Search trees and how they work<br>• Finding a node in a binary search tree<br>• Inserting a node<br>• Recursively traversing the tree in In order, Pre | See the chapter 12 in [CLRS] Thomas H. Cormen, Introduction to Algorithms |

| | | |
|---|---|---|
| | and Post order<br>• Applications of tree traversing in sorting | See the chapter 11.1 in Michael T. Goodrich, Data Structures & Algorithms in Java |
| 12-13 | • Deleting a node in a Binary Tree with all three cases<br>• Efficiency of Binary Trees<br>• Handling duplicate nodes in BST<br>• Applications of BST<br>• Coding a complete message<br>• Balanced and unbalanced trees<br>• The AVL trees Overview | See the chapter 11.3 in Michael T. Goodrich, Data Structures & Algorithms in Java |
| 14-15 | • Simple sorting<br>o Understanding why sorting is important<br>o Bubble sort<br>o Selection sort<br>o Insertion sort<br>o Merge Sort<br>o Quicksort<br>o Efficiency of Quicksort | See the chapter 7 in [CLRS] Thomas H. Cormen, Introduction to Algorithms<br><br>See the chapter 12 in Michael T. Goodrich, Data Structures & Algorithms in Java |
| 16 | • Hashing<br>• Applications of Hashing<br>• Direct Address<br>• Chain based Scheme<br>• Hash Tables | See the chapter 11 in [CLRS] Thomas H. Cormen, Introduction to Algorithms<br><br>See the chapter 10.2 in Michael T. Goodrich, Data Structures & Algorithms in Java |
| | • Graphs<br>o Introduction<br>o Searches (DFS & BFS) | See the chapter in Michael T. Goodrich, Data Structures & Algorithms in Java |

## Text Book

1. Introduction to Algorithms by Thomas H. Cormen, 3rd edition.
2. Data Structures & Algorithms in Java by Michael T. Goodrich, 6th edition.

## Reference Material

1. C++ plus Data Structures, 3rd edition, Nell Dale.
2. Think Data Structures: Algorithms and Information Retrieval in Java by Allen B. Downey

## Course Learning Outcomes

| | Course Learning Outcomes (CLO) |
|---|---|

| | |
|---|---|
| 1 | Demonstrate the knowledge of linear and non-linear data structures such as array, list, queue, stack, trees and graphs. |
| 2 | Implement algorithms for the efficient representation and manipulation of data. |
| 3 | Analyze different data structures to identify errors and predict the output. |

CLO-SO Map

| | SO IDs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLO ID | GA1 | GA2 | GA3 | GA4 | GA5 | GA6 | GA7 | GA8 | GA9 | GA10 | GA11 | GA12 |
| CLO 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CLO 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CLO 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Approvals

| | |
|---|---|
| Prepared By | Abdul Ghafoor |
| Approved By | Not Specified |
| Last Update | 19/8/2024 |

**Program Learning Outcomes**

**GA: Graduate Attributes**

**GA1 Computing Knowledge:** An ability to apply knowledge of mathematics, science, computing fundamentals and computing specialization to the solution of complex computing problems.

**GA2 Problem Analysis:** An ability to identify, formulate, research literature, and analyze complex computing problems reaching substantiated conclusions using first principles of mathematics, natural sciences and computing sciences.

**GA3 Design/Development of Solutions:** An ability to design solutions for complex computing problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

**GA4 Investigation:** An ability to investigate complex computing problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretationof experimental data, and synthesis of information to derive valid conclusion.

**GA5 Modern Tool Usage:** An ability to create, select and apply appropriate techniques, resources, and modern IT tools, including prediction and modeling, to complex

computing activities, with an understanding of the limitations.

**GA6 The Computer Scientist and Society:** An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional computing practice and solution to complexcomputing problems.

**GA7 Environment and Sustainability:** An ability to understand the impact of professional computing solutions in societal and environmental contexts and demonstrate knowledge of andneed for sustainable development.

**GA8 Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of computing practice.

**GA9 Individual and Team Work:** An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.

**GA10 Communication:** An ability to communicate effectively, orally as well as in writing, oncomplex computing activities with the computing community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**GA11 Project Management:** An ability to demonstrate management skills and apply computing principles to one's own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

**GA12 Lifelong Learning:** An ability to recognize importance of, and pursue lifelong learningin the broader context of innovation and technological developments