Nirma University
Institute of Technology

Computer Science and Engineering department

Course Policy

B.Tech. Computer Engineering / Information Technology

Semester: III, **Academic Year:** 2019-20, **Term:** ODD

Course Code & Name	:	2CS301 Data Structures
Credit Details	:	Lecture-3, Tutorial-0, Practicals-2 **Credits-4**
Course Co-ordinator	:	Prof. Jitali Patel
Contact No. & Email	:	07930642105, jitali.patel@nirmauni.ac.in
Office	:	N-F503
Visiting Hours	:	9:00 AM to 4:00 PM
Course Blog	:	https://ce403mkk.wordpress.com/
Course Faculty	:	**1. Prof. Jitali patel** **Visiting Hours**: Monday - 10.50 a.m. to 11.15 a.m. Odd Saturdays: 02:00 p.m. to 04:00 pm
2. Prof. Anitha Modi **Email:** anitha.modi@nirmauni.ac.in Office: N-F505 **Visiting Hours:** Monday - 10.50 a.m. to 11.15 a.m. Odd Saturdays: 02:00 p.m. to 04:00 pm	**3. Prof. Rupal Kapdi** **Email:** rupal.kapdi@nirmauni.ac.in Office: PG201 **Visiting Hours:** Monday - 10.50 a.m. to 11.15 a.m. Odd Saturdays: 02:00 p.m. to 04:00 pm	**4. Prof. Rajesh Pradhan** **Email:** rajesh.pradhan@nirmauni.ac.in Office: N-F504 **Visiting Hours:** Monday - 10.50 a.m. to 11.15 a.m. Odd Saturdays: 02:00 p.m. to 04:00 pm
Course Blog:	https://ce403mkk.wordpress.com/	

1. Introduction to Course:

- **Importance of course:**

  ✓ Data Structure is process through which we can collect and organize data in best way as well as perform operation on that in most effective way. If we have good understanding of data structures then we are specialized in organizing and storing data. This course covers the modern theory of algorithms, focusing on the themes of efficient algorithms and intractable problems. The course goal is to provide a solid background in algorithms for computer science students, in preparation either for a job in industry or for more advanced courses at the graduate level.

- **Course objective:**
  ✓ To develop students' knowledge in data structures and the associated algorithms. To introduce the concepts and techniques of structuring and operating on Abstract Data Types in problem solving.
  ✓ To discuss common sorting, searching and graph algorithms, and to study the complexity and comparisons among these various techniques.

Pre-requisite: Preliminary knowledge basic programming languages like C and C++ will help students to understand basic concepts of this subject.

2. Course Learning Outcomes(CLO):

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

CLO 1	:	Analyse various data structures and their applicability
CLO 2	:	Comprehend and Implement various techniques for searching and sorting
CLO 3	:	Identify the appropriate data structure to design efficient algorithm for the given application

3. Syllabus:

Syllabus:	**Teaching Hours:**
Unit I **Introduction to Data Structures:** Basic Terminology, Elementary Data Structure Organization, Classification of Data Structures: Primitive and Non-primitive, Linear and Non-linear, Operations on Data structures, Asymptotic notations, Notion of recursive algorithms.	**06**
Unit II **Linear Data Structures:** Introduction, variations, operations and applications of array, queue, stack and linked list	**12**
Unit III **Non Linear Data Structures:** Concepts and types of trees, tree traversal algorithms, search trees, Priority queue implementation and applications,	**12**

Representations of Graphs, Graph algorithms i.e. traversals, minimum spanning tree, shortest path, Traveling Salesman Problems

Unit IV **07**
Indexing structure: Concepts and implementations of B-Tree, B+ tree, Hashing, Dictionary

Unit IV **08**
Searching and Sorting Algorithms: Linear search, Binary search, internal and external sorting algorithms, sorting without comparison.

Suggested Readings^:

1. Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill
2. Tanenbaum, Data Structures using C & C++, PHI
3. Robert L. Kruse, Data Structures and Program Design in C, PHI
4. Mary E.S. Loomis, Data Management and file processing, PHI

L = Lecture, T = Tutorial, P = Practical, C = Credit

^ this is not an exhaustive list

Self-study:

Trees and Graphs: Dynamic Storage Management-Fixed Block Storage Allocation, First-Fit Storage Allocation, Storage Release Buddy System, Garbage Collection and Compaction

Searching Techniques: Weight Balance Trees, 2-3 Trees

Reference and Text books:

1. Jean-Paul Tremblay and Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill
2. Tanenbaum, Data Structures using C & C++, PHI
3. Robert L. Kruse, Data Structures and Program Design in C, PHI
4. Mary E.S. Loomis, Data Management and file processing, PHI

4. Laboratory details: (List of Experiments, Schedule, assessment policy)

Sr. No.	Topic	Hour (s)	Mapped CLO's
1.	**Array and Structures** a) Create a two dimensional array and perform insertion, deletion and update operation using pointer. b) Create an array of structure and perform insertion, deletion and update operation using pointers. c) Implement a matrix multiplication of two matrices stored in row major/ column major order.	**04**	**CLO1, CLO3**
2.	**Stack** a) Implement the basic stack operations using an array. b) Implement a program to convert fully parenthesized infix expression to postfix expression.	**04**	**CLO1, CLO3**
3.	**Queue** a) Implement circular queue with all operations b) Implement priority queue using an array.	**04**	**CLO1, CLO3**
4.	**Singly Linked List** a) Implement Traversal, Insertion and Deletion operations on singly linked list. b) Implement a program to reverse a singly linked list.*	**02**	**CLO1**
5.	**Doubly Linked List** a) Implement Traversal, Insertion and Deletion operations on doubly linked list. b) Implement an addition of two polynomial equations using linked list.	**04**	**CLO1, CLO3**
6.	**Sorting** a) Implement Quick/merge sort for sorting a given set of integers in ascending order. b) Implement Heap sort algorithm for sorting a given set of integers in ascending order. c) Implement merge sort for sorting a given set of integers in descending order. *	**04**	**CLO2**
7.	**Searching** Implement Binary search operation on a given set of integers.	**02**	**CLO2**
8.	**Binary Search Tree** Implement the following operations on given binary search tree :	**02**	**CLO3**

	Insert a node, Delete a node & Traverse the tree (Inorder, Preorder, Postorder)		
9.	**Tree** Implement AVL tree insertion, deletion and display operations.	**04**	**CLO3**
10.	**Graph** Implement an algorithm to obtain a spanning tree of a connected undirected graph using appropriate data structure.	**02**	**CLO1, CLO3**
	Total	**32**	

*** indicates extra exercises for practice**

5. Assessment Policy

5.1 Component wise Continuous Evaluation (CE), Laboratory and Project Work (LPW) & Semester End Examination (SEE) weightage

Assessment scheme	***CE***			***LPW***		***SEE***
Component weightage	*0.4*			*0.2*		*0.4*
	Class Test 30%	*Sessional Exam 40%*	*Conceptual Test 30%*	*Continuous Evaluation 75%*	*Viva Voce 25%*	

5.2 Assessment Policy for Continuous Evaluation (CE)

Assessment of Continuous Evaluation comprises of three components.

1. Class Test will be conducted as per academic calendar. It will be conducted online/ offline for the duration of 1 hour and will be of 30 marks.
2. Sessional Exam will be conducted as per academic calendar. It will be conducted offline for the duration of 1 hour and 15 minutes and will be of 40 marks.

6.3 Assessment Policy for Laboratory and Project Work (LPW)

Assessment of Laboratory and Project Work comprises of two components.

1. Continuous assessment for laboratory experiments will be conducted. There will be 10 experiments, each carrying weightage of 10 marks. At the end of

the course total marks obtained out of 100 will be converted according to weightage assigned. Assessment of Experiment will be carried out based on overall logic design and timely submission of practical.

2. A Viva voce examination for LPW component will be conducted as per academic calendar. It will carry a weightage of 25 marks.

6.4 Assessment Policy for Semester End Examination (SEE)

A written examination of 3 hour duration will be conducted for the course as per academic calendar. It will carry 100 marks and marks obtained out of 100 will be converted as per weightage assigned.

7. Lesson Plan

Sr. No.	Topic	Hour (s)	CLO	Applications
1	**Introduction to Data Structures**	[2]	1	Data structure used in various technical and real life application. Array is used to implement vectors, matrices, hash table, stack and queue.
	➢ Data Structures: -Types of Data Structures, Linear & non-linear Data Structures	1		
	➢ Array and storage structures for array	1		
2	**Different types of data structures:** **Stack**	[16]	1,3	**Stack:** Stack is used in undo\redo operation in word processors, Expression evaluation and syntax parsing **Queue:** Transport and operations research where various entities are stored and held to be processed. **Linked List:** In line editor, implementatio
	➢ Stack operations like PUSH, POP, PEEP, CHANGE	1		
	➢ Applications of stack			
	-Recursion	1		
	-Infix, Postfix & Prefix notations, Infix to Postfix conversion	2		
	- Evaluation of postfix expression	1		
	-Tower of Hanoi using stack	1		
	Queue			
	➢ Simple: Algorithms & Implementation of simple Queue	1		
	➢ Circular: Algorithms & Implementation of Circular Queue	1		
	➢ Doubly ended queue: Algorithms & Implementation of Circular Queue	2		
	Linked Lists	1		

	➢ Singly linked list: Insertion and Deletion algorithms ➢ Doubly linked lists: Insertion and Deletion algorithms ➢ Circular linked list: Insertion and Deletion algorithms ➢ Applications of linked list	2 2 1		n of sparse matrix and also to implement stack and queue
3	**Sorting algorithms** ➢ Insertion sort & Selection Sort ➢ Shell sort & Bubble sort ➢ Quick sort ➢ Heap sort ➢ Merge Sort ➢ Radix sort ➢ Tree sort	[6] 1 1 1 1 1 1	1,2	Sorting techniques are basic in any computing which makes many problem easy such as searching, frequency distribution, selection, closest pairs
4	**Searching techniques** ➢ Linear Search ➢ Binary Search ➢ Height Balanced Trees ➢ Weight Balance Trees, 2-3 Trees(self-study) ➢ Hashing techniques	[6] 1 1 2 2	1,2	Searching techniques are used to search the required data in stack, queue, linked list and also finding a node in a tree.
5	**Trees & Graphs** ➢ Definitions of Trees & graphs ➢ Creation of Simple Binary tree ➢ Conversion of General Tree in to Binary tree ➢ Threaded representation of tree ➢ Insertion, Deletion and Traversal of binary search tree ➢ Graph structure & Representation of graph ➢ Traversing of Graph (DFS &BFS) ➢ Applications of graph ➢ Dynamic Storage Management-Fixed Block Storage Allocation, First-Fit Storage Allocation, Storage Release Buddy System, Garbage Collection, Compaction(self-study)	[11] 1 1 1 1 3 1 2 1	1,3	**Trees:** Trees are used in Parsers, Filesystem, to represent network topology as a tree **Graphs**: Graph are used in Connections/relations in social networking sites, Routing ,networks of communication
6	**Indexing Structure** ➢ Concepts and implementations of B-Tree ➢ B-Trees and B+ tree ➢ Dictionary	[4] 1 2 1	1,3	used in database management systems

	Total	**45**		

Note: Topics marked as red are for self-study.

8. Mapping of Session Learning Outcomes (SLO) with Course Learning Outcomes (CLO)

Session No.	**Session Learning Outcomes:** **After successful completion of the session, student will be able to**	**CLO**
1.	Understand importance, scope and policy of the course Understand the need of various data structures.	-
2.	Prerequisite : Pointer, Structure and function	-
3.	Prerequisite: Asymptotic notation: Time and Space Complexity.	-
4.	Understand the fundamentals of array data storage in 1D, 2D and ND.	1
5.	Ability to calculate physical address for the array data storage in Row Major and Column major formats:	1
6.	Understand stack data structure and its basic operations. Identify applicability of stack data structure	1,3
7.	Understand recursive process and applications where recursion is used.	3
8.	Understand the process of conversion from infix to postfix. Apply the knowledge and convert expressions in suggested formats	3
9.	Understand the process of conversion from infix to prefix. Apply the knowledge and convert expressions in suggested formats	3
10.	Understand the process of evaluating a given expression. Ability to solve expressions in given formats	3
11.	Ability to trace recursive applications such as tower of Hanoi.	3
12.	Understand queue data structure and its basic operations. Identify applicability of queue data structure	1
13.	Understand circular queue data structure and its basic operations.	1
14.	Understand double ended simple queue data structure and its basic operations.	1
15.	Understand Priority queue data structure and its basic operations.	1,3
16.	Understand dynamic data structures and their applicability. Understand singly linked list and its basic operations and their algorithms.	1
17.	Understand doubly linked list and its basic operations	1

18.	Ability to write algorithms to basic operations of doubly linked list.	1
19.	Understand circular doubly linked list and its basic operations	1
20.	Ability to write algorithms to basic operations of circular doubly linked list.	1
21.	Ability to Identify areas where linked list is applicable.	3
22.	Understand tree construction process. Ability to implement the basic structure.	1
23.	Understand height balanced trees and their data structure with applications	1,3
24.	Understand hashing technique and their real world application	1,3
25.	Ability to resolve hash collisions and calculate solutions to given hashing problem.	1,3
26.	Understand tree and graph structure and their real world implementations and applications.	1,3
27.	Ability to create Simple Binary tree	1
28.	Ability to convert General Tree in to Binary tree	1
29.	Understand binary search tree its construction and application	1,3
30.	Understand the basic binary search tree operations.	1
31.	Ability to implement basic binary search tree operations and traversals.	2
32.	Understand Graph structure & Representation of graph	1
33.	Understand DFS traversal technique Ability to solve problems of traversals	1,3
34.	Understand BFS traversal technique Ability to solve problems of traversals. Travelling Salesman problem.	1,3
35.	Understand the Implementation and Traders of Sequential Access, B-Trees,	3
36.	Minimum Spanning Tree: Prim’s and Kruskal’s Algorithm	3
37.	Understand B-tree and solve problem related to B-trees	1,2
38.	Understand the sorting techniques and trace insertion and selection sort.	1,2
39.	Understand the bubble and shell sorting techniques Ability to trace bubble and shell sorting techniques.	2
40.	Understand the quick sorting techniques Ability to trace quick sorting techniques.	2
41.	Understand the heap sorting techniques Ability to trace heap sorting techniques.	2
42.	Understand the merge sorting techniques Ability to trace merge sorting techniques.	2
43.	Understand the radix sorting techniques Ability to trace radix sorting techniques.	2
44.	Understand the radix sorting techniques Ability to trace radix sorting techniques.	2
45.	Understand the searching techniques. Ability to trace linear and binary searching techniques.	2

9. Teaching-learning methodology

1. Lectures: Primarily Chalk and Black board, Power Point Presentations (PPTs) and Demonstration of concepts through web pages will be used to conduct the course. However, where required, Video Lectures, Animations etc. will be used to enhance the teaching-learning process.
2. Laboratory: Explanation of Experiment to be performed along with co-relation with theory will be given. At the end of each session assessment will be carried out based on parameters various tags utilization, overall design of web page and logic (in case of JavaScript) and timely submission of practical.

10. Active learning techniques

Active learning is a method of learning in which students are actively or experientially involved in the learning process. Following active learning techniques will be adopted for the course.

1. Recall, Summarize, Question, Connect, and Comment: At the beginning of class, students are asked to recall and list the most important points from the previous class. They then summarize these points in sentences. Next students write one questions from the previous material that they wanted answered. Fourth, they are instructed to make one connection between what they learned in the previous class and any of the classes before that. Finally, they are asked to comment on how confident they felt.
2. Flipped Class-room: In the flipped classroom, instructors assign video lectures or reading material as homework, and use class time for active learning exercises and direct engagement with students.

11. Course Material

Following course material is uploaded on the course website:

https://sites.google.com/a/nirmauni.ac.in/2cs301-data-structures-and-algorithms/

Course Policy

- Lecture Notes
- Books / Reference Books / NPTEL video lectures
- Assignments, Lab Manuals
- Question bank
- Web-links, Blogs, Video Lectures, Journals
- Softwares
- Advanced topics

12. Course Learning Outcome Attainment

Following means will be used to assess attainment of course learning outcomes.

- Use of formal evaluation components of continuous evaluation, laboratory work, semester end examination
- Informal feedback during course conduction

13. Academic Integrity Statement

Students are expected to carry out assigned work under Continuous Evaluation (CE) component and LPW component independently. Copying in any form is not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Turnitin software will be used to check plagiarism wherever applicable. Academic integrity is expected from students in all components of course assessment.