LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Design (703)

Bachelor of Engineering (B.E.) – Semester - II

Course Code:	117033292
Course Name:	**Data Structures using Java**
Category of Course:	Professional Core Course (PCC)
Prerequisite Course:	Computer Programming using Java – I (117032191)

Teaching Scheme				
Lecture (L)	Tutorial (T)	Practical (P)	Credit	Total Hours
5	0	2	6	30

Syllabus				
Unit No.	**Topic**	**Prerequisite Topic**	**Successive Topic**	**Teaching Hours**
01	**Introduction to Data Structure**			**2 (7%)**
	1.1 Data Types – Primitive and Non-Primitive	Computer Programming using Java – I (117032191 Unit-2.4)		
	1.2 Types of Data Structures- Linear & Non-Linear Data Structures	---		
	1.3 Memory Representation of Array: Row Major and Column Major	Arrays (117032191 - Unit-5.1, 5.2)		
02	**Stack-1**			**3 (10%)**
	2.1 Stack-Definitions & Concepts, Operations on Stacks (Push, Pop, Peep, Change)	Arrays (117032191 - Unit-5.1, 5.2)		
	2.2 Applications of Stacks: Tower Hanoi Using recursion, Reverse the String using Stack, Decimal to Binary conversion using Stack	Stack-1(117033292-Unit-2.1)	Recursively Enumerable Languages (017033492-Unit-10.1)	
03	**Stack-2**			**4 (13%)**
	3.1 Applications of Stacks: Expression Conversion (Infix, Polish, Reverse-Polish)	Stack-1(117033292-Unit-2.1)		
	3.2 Applications of Stacks: Expression Evaluation and Implementation (Prefix, Postfix)	Stack-1(117033292-Unit-2.1)		
04	**Queue-1**			**3 (10%)**
	4.1 Queue - Definitions & Concepts, Representation of Queue, Operations on Simple Queue (Enqueue, Dequeue, Display)	Arrays (117032191 - Unit-5.1, 5.2)	Process Management (017033301 -Unit-2.1,2.2,2.3)	
	4.2 Operations on Circular Queue (Enqueue, Dequeue, Display)	Queue-1(117033292 -Unit-4.1)		
05	**Queue-2**			**2 (7%)**
	5.1 Priority Queue - Definition and Introduction and Circular Double Ended Queue – Introduction, Types (Input restricted, Output restricted), Operations on Circular double ended queue – Insert at Front, Insert at Rear, Delete at Front, Delete at Rear	Queue-1(117033292 -Unit-4.1)		
	5.2 Applications of Queue	---		
06	**Singly Linked List**			**3 (10%)**
	6.1 Dynamic Memory Allocation, Memory Representation of Linked List, Availability Stack, Types of Linked List	Object Fundamentals (117032191 - Unit-7.1)	File System (017033301 - Unit-8.2)	
	6.2 Operations on Singly Linked List (Insert, Delete, Traverse)	---		
	6.3 Implementation of Stack and Queue using Singly Linked List	Stack-1(117033292-Unit-2.1), Queue-1(117033292-Unit-4.1), Singly Linked List (117033292-Unit-6.2)		
07	**Doubly and Circular Link List**			**3 (10%)**
	7.1 Operations on Doubly Linked List (Insert, Delete, Traverse)	Singly Linked List (117033292-Unit-6.2)		
	7.2 Operations on Circular Linked List (Insert, Delete, Traverse)	Singly Linked List (117033292-Unit-6.2)		
	7.3 Applications of Linked List	---		
08	**Tree**			**4 (13%)**
	8.1 Tree-Definitions & Concepts - Root, Parent, Child, Leaf, Ancestor of Node, Descendant of Node, Siblings, Level of Node, Internal Node, Height of Node and Tree, Depth of Node and Tree, Subtree, Neighbor of a Nodes, Types of Tree (General Tree, Binary Tree, Balanced Tree, Full Binary Tree, Complete Binary Tree, Perfect Binary Tree, Skewed Tree, Degenerate or pathological Tree	---	Sorting Techniques (017033591 -Unit-2.4)	
	8.2 Representation of Binary Tree using arrays and LinkedList, Binary Tree Traversal (Inorder, Postorder, Preorder), Construction of Binary Tree from Traversal	Tree (117033292-Unit-8.1)		
	8.3 Conversion from General Tree to Binary Tree, Threaded Binary Tree and its types - Left Threaded, Right Threaded, Full Threaded Binary Tree, Expression Tree	Tree (117033292-Unit-8.2)		

	8.4 Introduction to Heap and its Types - Min Heap and Max Heap, Operations on heap - Insertion, Deletion	Tree (117033292-Unit-8.2)	Sorting Techniques (017033591 -Unit-2.4)	
	Applications of Trees			
09	9.1 Binary Search Tree: Definition and Implementation, Operations: Search, Insertion, Min, Max, Successor, Predecessor, Deletion	Tree (117033292-Unit-8.2)		**4 (13%)**
	9.2 Introduction to Height Balanced BSTs: AVL and Balance Mechanism, Operations on AVL trees - Insertion, Deletions and Rotations	Tree (117033292-Unit-8.1)		
	9.3 Multi-way Search Tree: 2-3 Trees (Order 3 B Tree), B and B+ tree , Insertion Operation	Tree (117033292-Unit-8.1)	Indexing (017033291 -Unit-7.2)	
	Hashing			
10	10.1 Hashing: Introduction: Hashing, Symbol Table, Hashing Functions by using division method	---	File System (017033301 - Unit-8.2)	**2 (7%)**
	10.2 Collision Resolution Techniques - Chaining, Linear Probing, Quadratic probing, Double hashing	---		
	10.3 Applications of Hashing	---		

Sr No.	Practical Title	Link to Theory Syllabus
1	Write a program to calculate the address of a 1- D array.	Unit-1
2	Write a program to calculate the address of array in 2 – D array. Where the type of an array – Row major order or column major order – base address and the index where we want to find the address is given as inputs.	Unit-1
3	Write a program to implement stack operations using an array (Operations: push, pop, peep, change, Display)	Unit-2
4	Write a program to reverse the string using stack.	Unit-2
5	Write a program to evaluate the given postfix expression.	Unit-3
6	Write a program to implement simple queue insertion and deletion using an array	Unit-4
7	Write a program to implement circular queue insertion and deletion using an array	Unit-4
8	Write a program to implement DEQUEUE to insert a node using an array	Unit-5
9	Write a program to implement DEQUEUE to delete a node using an array	Unit-5
10	Write a program to create a singly linked list of three nodes where nodes are inserted at the front.	Unit-6
11	Write a program to create a singly linked list of three nodes where nodes are inserted at the last.	Unit-6
12	Write a program to delete a node at particular position from a given singly linked list.	Unit-6
13	Write a program to implement insertion in circular Linked list	Unit-7
14	Write a program to implement deletion in circular Linked list	Unit-7
15	Write a program to implement insertion in doubly Linked list	Unit-7
16	Write a program to implement deletion in doubly Linked list	Unit-7
17	Write a Program to Find the Height of a Binary Tree	Unit-8
18	Write a Program to Implement in – order, pre – order and post – order traversal in a given Binary Tree.	Unit-8

Major Components/ Equipment

Sr. No.	Component/Equipment
1	Computer
2	Notepad++, VS Code

Proposed Theory + Practical Evaluation Scheme by Academicians (% Weightage Category Wise and it's Marks Distribution)					
L:	**5**	**T:**	**0**	**P:**	**2**

Note: In Theory Group, Total 4 Test (T1+T2+T3+T4) will be conducted for each subject.
Each Test will be of 25 Marks.
Each Test Syllabus Weightage: Range should be 20% - 30%

Group (Theory or Practical)	Group (Theory or Practical) Credit	Total Subject Credit	Category	% Weightage	Marks Weightage
Theory			MCQ	23%	45
Theory			Theory Descriptive (Mainly Programming)	12%	25
Theory	**5**	**6**	Formulas and Derivation	0%	0
Theory			Numerical	15%	30
Expected Theory %	**50%**		**Calculated Theory %**	**50%**	**100**

Practical			Individual Project	25%	50
Practical			Group Project	15%	30
Practical	1		Internal Practical Evaluation (IPE)	10%	20
Practical			Viva	0%	0
Practical			Seminar	0%	0
Expected Practical %	**50%**		**Calculated Practical %**	**50%**	**100**
Overall %	**100%**			**100%**	**200**

Course Outcome	
	Upon completion of the course students will be able to
CO1	Understand the concept of memory management, basic data structures with its types and fundamental data structure stack to develop problem-solving skills based on stack that is applicable across various computational domains.
CO2	Understand and utilize queue and LinkedList data structures, including various types and its operations, enhancing their problem-solving skills across diverse computational scenarios. Understand the concept of static versus dynamic data structures.
CO3	Master diverse LinkedList and tree data structures with its operations. Through practical applications, students will understand the versatility of linked lists and trees in different complex problems
CO4	Possess advanced skills in managing different Tree data structure and its operations. Understand the importance of trees in searching. Furthermore, students will grasp the concepts of hashing along with their practical applications
Suggested Reference Books	
1	An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill.
2	Data Structures: A Pseudo-code approach with C -By Gilberg & Forouzan Publisher Thomson Learning.

List of Open Source Software/Learning website	
1	www.geeksforgeeks.org
2	www.tutorialspoint.com
3	www.programiz.com
4	https://visualgo.net/en

Practical Project/Hands On Project		
Sr. No.	**List of Practical Projects**	**Linked with Unit**
1	Hotel management system : Listed below are some of the important functions **dashboard()** – This function displays the menu or welcome screen to perform different hotel booking activities mentioned below. **new_acc()** – This function creates a new customer account. It asks for some personal and banking details of the customer such as name, date of birth, citizenship number, address and phone number. **room_type()** – This function allows the user to select the categories of the room ie normal or executive with the option of Ac room or non ac room. **check_availability()** – This functionality allows the user to check the number of room vacant prior booking. **book_room()** – This function allows the user to book the selected room. **search_facilities()** – With this function, if the user selects the executive room than user can search for the extra facilities provided like games, swimming, food service in rooms while booking. **payment()** – This function allows making payment of booked room based on number of days the room is occupied via online method option or at the checkout time.	Unit 2,3,4,6
2	It is required to maintain and process the status of total 9 resources. The status value is to be stored in an integer array of dimension 3x3. The valid status of a resource can be one of the 3 followings: **free**: indicated by integer value 0 **occupied**: indicated by integer value 1 **inaccessible**: indicated by integer value 2 Declare a class called ResourcesStatus, having data member called statusRef, referring to a two dimensional array (3x3) of integers to be used to refer to the above mentioned status values. Define a member method called processStausCount that counts and displays total number of free resources, total number of occupied resources and total number of inaccessible resources. The exception to be raised and handled if total number of occupied resources exceeds total number of free resources. The handler marks status of all inaccessible resources as free. Accept initial status values from user and initialize the array. Raise and handle user defined exception if invalid status value given.	Unit 2,3,4,6,8
3	Create an application that performs the following task associated with the files : 1) Eliminating repeated lines from the files. 2) Reverse the content of file and store in another file. 3) Remove the lines starting from any prefix. 4) Obtain the line number where the particular word is present. 5) Obtaining number of words, characters, white spaces and lines present in that particular file.	Unit 2,3,4,5,6
4	Implement calculator functionality.	All
5	Write a program to implement Quadratic equation.	All