Department of MCA

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Data Structures Using C
2.	Course Code	BCAMJ-201
3.	Academic Year	2023-24
4.	Semester	2nd
5.	Number of Lesson plans	45
6.	Faculty Assigned	Tajamul Hassan

Tajamul Hassan

Faculty Signature

Lesson Plan No. 1.1	Course Name: Data Structure using C Topic: Number Systems	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Articulate the fundamental concept of Number System. b. Identify different types of number Systems. c. Convert one number system to another.
Teaching Aids (if any)	a. Projector, Slides b. Use of Near pod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What do you understand by number system? What type of number system do you use for general purpose? - Introduce the concept of number system - Talk about the need of different number systems 2. **Development** (30 minutes) a. Concept of different number systems - Introduce different number systems - Talk about Decimal, Binary, Octal, HexaDecimal b. Conversion of different number systems - Conversion of decimal to binary and binary to decimal. - Conversion of decimal to octal, octal to decimal - Conversion of octal to binary and binary to octal - Conversion of decimal to hexadecimal and hexadecimal to decimal. - Conversion of hexadecimal to binary and binary to hexadecimal.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on number system. https://nptel.ac.in/courses/106105085 3. Homework - Convert the given numbers to another forms and submit the answers on Google classroom Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.2	Course Name: Data Structure using C Topic: Introduction to Data Structures	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Articulate the fundamental concept of Data Structures. b. Understand the need of data Structure. c. Identify different types of data structures. d. Operations performed on data structures.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Ask questions - What do you understand by the word data? - What are the different types of data present in C language? - Do you know why is it important to store data in a particular manner? - Introduce the definition of Data Structures. - Talk about thereal world examples of data structures. - Highlight the important characteristics of data structures. 2. **Development** (30 minutes) a. Data Structure basics - Introduce the concept of data structure including its commonly used applications like plotting of graphs, text editing etc. b. Need of data structures - Processor speed - Data Search - Multiple request c. Classification of data structures - Introducing primitive data structures(built-in data types) and non-primitive data structures(derived data types) -Show flow chart includingexamples of primitive data structure (int,char,floatetc) and non primitive data structure(array, linked list, stack,queue , graphs,trees). - Give overview of non-primitive data structure. d. Need of data structures - Processor speed - Data Search - Multiple request

	e. Basic Operations performed on Data Structure - Traversing - Insertion - Deletion - Sorting
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 3. Homework - Create a presentation highlightingdata structure concepts and submit on Google classroom Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Near pod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.3	Course Name: Data Structure using C Topic: Arrays	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Articulate the fundamental concept of Arrays. b. Understand the need of Arrays. c. Identify different types of Arrays. d. Operations performed on Array.
Teaching Aids (if any)	a. Projector ,Slides b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What do you understand by Array? What is the use of Array? How does the array store Data? - Introduce the definition of Array. - Talk about theexamples of Array. - Highlight the different types of Array. 2. **Development** (30 minutes) a. Array basics - Declaration of Array - Initializing Array - Accessing elements of Array b. Need of Arrays - Stores multiple number of elements. - Stores any kind of primitive data type. c. Classification of Arrays - 1-D Array - Multi-Dimensional Array d. Basic Operations performed on Arrays - Traversing - Insertion - Deletion - Sorting - Searching

	e. Concept of Searching in Arrays - Linear Search(search an element or value in a given array by traversing the array from the starting, till the desired element or value is found). - Binary Search(Searching a sorted array by repeatedly dividing the search interval in half.) **3.** Exercise(5 minutes) - Ask Students to write down the program for Searching elements in an array. - Asking them which technique is better.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.4	Course Name: Data Structure using C Topic: Multidimensional Arrays	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the different sorting techniques. b. Articulate basic concepts ofMulti-Dimensional Arrays. c. Perform manipulations on Arrays
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions What are the different operations that can be performed on array? What is Linear Search? What is Binary Search? - Introduce the concept of sorting. - Highlight the different types of Sorting techniques. **2. Development** (30 minutes) a. Sorting of array (ascending/descending order) b. Different sorting techniques -Bubble Sort -Selection Sort -Insertion Sort -Merge Sort -Quicksort -Radix Sort - Shell sort c. Introduction to multi-dimensional array - Declaration - Initialization - Accessing elements in 2-D array d. Manipulations on 2-D array - Addition

	- Subtraction - Multiplication **3.** Exercise(5 minutes) - Write down a program to add rows of 2-D array - Write down a program to add columns of 2-D array
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.5	Course Name: Data Structure using C Topic: Pointers and Arrays	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Articulate the fundamentalconcept ofPointers. b. Understand Pointers and Array. c. Advantages of pointer.
Teaching Aids (if any)	a. Projector ,Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions How to initialize 2-D array? What is pointer? - Introduce the definition of pointer. - Give an example of pointer. **2. Development** (30 minutes) a. Pointer basics - Discussion of pointer with example. - Declaration of pointer. - A detailed example of pointers b. Pointers and array - Relation of arrays and pointers. c. Advantages of pointer - Reduces the code and improves the performance. - Return multiple values from a function. - access any memory location d. Uses of Pointers -Dynamic memory allocation - Arrays, Functions, and Structures **3.** Exercise(5 minutes) - Ask Students to write down the program to find the sum of n

	numbers with arrays and pointers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings.
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.6	Course Name: Data Structures using C Topic: De-referencing and Void Pointers	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of De-referencing. b. Know the need of De-referencing c. Articulate the basics of Void pointers.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions What do you understand by pointer? What is the output of the program (from pointers and array) displayed on the screen? - Introduce the concept of de-referencing. **2. Development** (30 minutes) a. De- Referencing basics - Representation of de-referencing pointer. - Steps to de-reference a pointer. - Detailed Example b. Need of dereferencing a pointer - to access or manipulate the datastored at the memory location, which is pointed by the pointer. - operation applied will directly affect the value of the variable that it points to. c. Basics of Void Pointers - Definition - Example of Void pointer d. Advantages of Void Pointer -Implement generic functions in C -malloc() and calloc() return void * type and this can help allocate memory of any data type

	3. Exercise(5 minutes) - Ask Students to write down the program giving example of void pointer.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.7	Course Name: Data Structures Using C Topic: Dynamic Memory Allocation	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Dynamic memory allocation. b. Identify and understand the basic functions used for dynamic memory allocation.
Teaching Aids (if any)	a. Projector,Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions What is De-Referencing? What is the function of void pointer? - Introduce the need of Dynamic memory allocation. **2. Development** (30 minutes) a. Dynamic memory allocation basics - definition of Dynamic memory allocation b. Library functions for Dynamic memory allocation -malloc(memory allocation) -calloc(contiguous allocation) -free(de-allocate) -realloc(re-allocation) c. Examples of all the functions - Difference between all the library functions (malloc,calloc, free,realloc) with the help of examples. **3. Exercise**(5 minutes) -Ask Students to identify the use of each function required in dynamic memory allocation. -Collect the responses using Nearpod.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.8	Course Name: Data Structures Using C Topic: Structures	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Articulate the concept of Structures. b. Understanding pointer to structures.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz.
Teaching Development	**1. Introduction** (5 minutes) - Ask questions Why is the need of dynamic memory allocation ? What are the different library function used in dynamic memory allocation? What are the function of- calloc(), malloc(), re-alloc(), free() - Introduce the basic concept of data structures **2. Development** (30 minutes) a. Structure basics - Defining a Structure - Accessing Structure Member - A well defined example b. Structure to pointer - Definition - Example of structure pointer. **3. Exercise**(5 minutes) - Ask Students to determine output of the program shown on the screen. - Collect the responses using Nearpod.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085

	Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.1	Course Name: Data Structure using C Topic: Introduction to Stacks	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the stack data structure b. Gain knowledge about different types of basic operations that can be performed in a stack. c. Articulate the applications of stack.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) Ask questions - What is the difference between arrays and linked lists? - What is the importance of both? - What is the difference between physical and logical data structures? - Introduce the concept of stacks **2. Development** (30 minutes) a. Introduction to stacks. - Examples of stacks in daily life. - Significance of top of the stack. b. Basic operations on stacks - push - pop - peek - isEmpty - isFull c. Applications of stacks - String reversal - UNDO/REDO - Recursion - DFS(Depth First Search) - Backtracking - Expression conversion - Memory management

	3. Exercise(5 minutes) - Ask students about theinsertion and deletion onto a stack
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.2.	Course Name: Data Structure using C Topic: Operations on Stacks with Implementation	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the different types of implementations of stacks. b. Gain knowledge about implementation of stacks in C using arrays c. Articulate about hoe to perform different stack operations using arrays.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) Ask questions - What is a stack data structure? - What is the importance static and dynamic stack implementation? - Introduce theconcept of implementation of stacks. **2. Development** (30 minutes) a. Introduction to arrays and stacks. - Significance of using arrays for implementing stack. - Explaining the array implementation of stack. b. Implementation of basic operations on stacks using arrays - push - pop - peek - isEmpty - isFull **3.** Exercise(5 minutes) - Ask students about theinsertion and deletion onto a stack - Stack A has the entries a,b,c(with a on top). Stack B is empty. An entry popped out of stack A can be printed immediately or pushed to stack B. An entry popped out of stack B can be only printed. In this arrangement, which of the following are not possible? b,a,c b,c,a

	c,a,b a,b,c
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.3.	Course Name: Data Structure using C Topic: Infix, Prefix and Postfix Expressions	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the different types of notations for expressions. b. Gain knowledge about parenthesis checking. c. Articulate about the conversion of infix to postfix expression.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions - What are the applications of stacks? - Why do we need to check the balancing of parenthesis? - Why do we need to convert infix into postfix expression? - Introduce theconcept of infix, prefix and postfix expressions. **2. Development** (30 minutes) a. Introduction to types of expressions with examples. - Infix - Postfix - Prefix b. Briefing the rules that are to be followed for the conversion of infix expression to the postfix expression. Explaining with examples. **3.** Exercise(5 minutes) - Ask studentsto evaluate the postfix expression for the following A+b*c A-b/c*d+e K + L - M*N + (O^P) * W/U/V * T + Q A-B+(M^N)*(O+P)-Q/R^S*T+Z
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

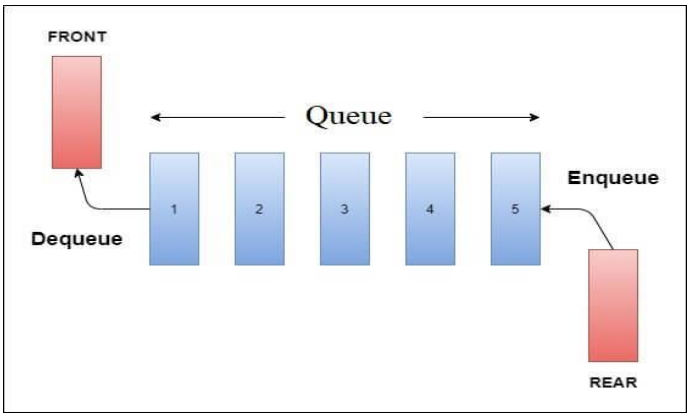
	2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.4.	Course Name: Data Structure using C Topic: Postfix Evaluation, Infix to prefix Conversion	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of postfix evaluation. b. Gain knowledge about the infix to prefix conversion. c. Articulate about how to evaluate an expression given in postfix notation.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) Ask questions - What is the difference between postfix and prefix notation? - Why do we need to perform the conversions? - Introduce the concept of postfix expression evaluation and infix-prefix conversion. **2. Development** (30 minutes) a. Introduction to postfix evaluation - Rules to evaluate a postfix expression. - Examples for evaluation 456*+ 53+83-* 35*62/+4- 2 3 1 * + 9 – 53+62/*35*+ b. Infix to prefix conversion - discuss the rules for infix to prefix conversion. **3.** Exercise(5 minutes) - Ask studentsto convert the given infix expression to prefix (A+B)+C-(D-E)^F
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085

	Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No.2.5	Course Name: Data Structure using C Topic: Queue	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the queue data structure b. Gain knowledge about different types of basic operations that can be performed in a queue. c. Articulate the applications of queue.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) Ask questions - What is the principle of working of a stack? - What is the importance of top variable? - What is the difference between insertion and deletion in terms of top variable? - Introduce the concept of queues. **2. Development** (30 minutes) a. Introduction to queues. - Examples of queues in daily life. - Significance of maintaining two variables in case of queue. • Front • Rear  b. Basic operations on stacks - enqueue - dequeue - peek

	- display - isEmpty - isFull c. Applications of stacks - Managing requests on a single shared resource - Handling hardware or real-time systems interrupts - Handling website traffic - Routers and switches in networking - Maintaining the playlist in media players **3.** Exercise(5 minutes) - Ask students about theinsertion and deletion in a queue
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 https://www.geeksforgeeks.org/queue-data-structure/ Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.6	Course Name: Data Structure using C Topic: Implementing the Pop Operation (Stack)	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the pop operation in the context of a stack data structure. b. Implement the pop operation using arrays and linked lists. c. Analyze the edge cases and potential errors in the pop operation.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for pop operation in stacks
Teaching Development	**1. Introduction** (5 minutes) - Recap the concept of stacks and their LIFO (Last In, First Out) principle. - Ask: What happens when we try to remove an element from a stack? - Introduce the pop operation as the mechanism for removing elements from a stack. **2. Development** (30 minutes) a. **Pop Operation Overview** - Explain the logic behind the pop operation. - Discuss the steps involved: checking for an empty stack, retrieving the top element, and updating the stack pointer. b. **Implementation Using Arrays** - Provide a code example of implementing the pop operation using an array-based stack. - Discuss potential issues such as stack underflow. c. **Implementation Using Linked Lists** - Provide a code example of implementing the pop operation using a linked list-based stack. - Compare the array and linked list implementations. **3. Exercise (5 minutes)** - Write code to implement the pop operation in a stack and handle

	edge cases. -
Closure	Summarize the importance of the pop operation and how it is implemented in different stack structures. • Suggested Reading "Data Structures and Algorithms in C" by Adam Drozdek
Evaluation	• Reflective Questions: What are the key steps in the pop operation? How does the implementation differ between arrays and linked lists? • Quick quiz on stack operations, focusing on pop.

Lesson Plan No. 2.7	Course Name: Data Structure using C Topic: Implementing the Push Operation (Stack)	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the push operation in a stack. b. Implement the push operation using arrays and linked lists. c. Handle stack overflow and other edge cases in the push operation.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for push operation in stacks
Teaching Development	**1. Introduction** (5 minutes) - Recap the structure and purpose of a stack. - Ask: How do we add elements to a stack? - Introduce the push operation and its significance in stack operations. **2. Development** (30 minutes) a. **Push Operation Overview** - Explain the logic of the push operation. - Discuss the process: checking for stack overflow, adding the element, and updating the stack pointer. b. **Implementation Using Arrays** - Provide a code example of implementing the push operation using an array-based stack. - Discuss handling of stack overflow. c. **Implementation Using Linked Lists** - Provide a code example of implementing the push operation using a linked list-based stack. - Compare the two implementations in terms of efficiency and complexity. **3. Exercise (5 minutes)** - Write code to implement the push operation and handle edge

	cases like stack overflow.
Closure	Summarize the push operation and its role in stack manipulation. • Suggested Reading "Data Structures Using C" by Reema Thareja
Evaluation	• Reflective Questions: What are the potential issues when implementing the push operation? How can these be handled? • Quick quiz on stack operations, focusing on push.

Lesson Plan No. 2.8	Course Name: Data Structure using C Topic: Multiple Queues	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of multiple queues and their applications. b. Implement multiple queues using array and linked list data structures. c. Analyze the scenarios where multiple queues are beneficial.
Teaching Aids (if any)	a. Projector, Slides b. Visual aids to demonstrate multiple queues
Teaching Development	**1. Introduction** (5 minutes) - Discuss the basic concept of a queue. - Ask: How would multiple queues be useful in a system? - Introduce the concept of multiple queues and their applications in operating systems, scheduling, and more. **2. Development** (30 minutes) a. **Multiple Queues Overview** - Explain the structure and use cases of multiple queues. b. **Implementation Using Arrays** - Provide code examples for implementing multiple queues using arrays. - Discuss the challenges and limitations. c. **Implementation Using Linked Lists** - Provide code examples for implementing multiple queues using linked lists. - Compare the array and linked list implementations in terms of efficiency and complexity. **3. Exercise (5 minutes)** - Implement a simple system with multiple queues and perform operations on them.

Closure	Recap the importance of multiple queues and their implementation methods. • Suggested Reading "Operating Systems: Internals and Design Principles" by William Stallings
Evaluation	• Reflective Questions: In what scenarios are multiple queues more effective than a single queue? • Quick quiz on the implementation and applications of multiple queues.

Lesson Plan No. 2.9	Course Name: Data Structure using C Topic: Priority Queues	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept and functionality of priority queues. b. Implement priority queues using arrays and heaps. c. Analyze the time complexity of various operations in a priority queue.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for priority queue operations
Teaching Development	**1. Introduction** (5 minutes) - Discuss the concept of a queue and its FIFO behavior. - Ask: What if some elements are more important than others? - Introduce the concept of priority queues and their importance in scheduling and other applications. **2. Development** (30 minutes) a. **Priority Queue Overview** - Explain the structure and purpose of priority queues. - Discuss different types of priority queues (max-priority and min-priority). - b. **Implementation Using Arrays** - Provide code examples for implementing priority queues using arrays. - Discuss the time complexity of insertion, deletion, and retrieval operations. - c. **Implementation Using Heaps** - Explain the use of heaps for efficient priority queue operations. - Provide code examples for heap-based priority queues. **3. Exercise (5 minutes)** - Implement a priority queue using both array and heap methods.
Closure	Summarize the lesson by discussing the significance of priority queues in

	various applications. • Suggested Reading "Data Structures and Algorithm Analysis in C" by Mark Allen Weiss
Evaluation	• Reflective Questions: What is the advantage of using a heap for implementing a priority queue? • Quick quiz on priority queue operations and their complexities.

Lesson Plan No. 2.10	Course Name: Data Structure using C Topic: Circular Queues	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept and implementation of circular queues. b. Implement circular queues using arrays. c. Analyze the advantages of circular queues over linear queues.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for priority queue operations
Teaching Development	**1. Introduction** (5 minutes) - Recap the structure and limitations of linear queues. - Ask: What happens when a linear queue reaches its end? - Introduce the concept of circular queues as a solution to the limitations of linear queues. **2. Development** (30 minutes) a. . **Circular Queue Overview** - Explain the structure and working of circular queues. - Discuss the advantages of circular queues over linear queues. b. **Implementation Using Arrays** - Provide code examples for implementing circular queues using arrays. - Explain the wrap-around behavior and handling of front and rear pointers. - Discuss potential issues like queue overflow and underflow. c. **Comparison with Linear Queues** - Analyze the scenarios where circular queues are more efficient than linear queues. **3. Exercise (5 minutes)** - Implement a circular queue and perform basic operations like enqueue and dequeue.

Closure	Recap the benefits of circular queues and their applications in real-world scenarios. □ Suggested Reading "Fundamentals of Data Structures in C" by Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed
Evaluation	□ Reflective Questions: Why are circular queues more efficient in some cases compared to linear queues? □ Quick quiz on circular queue operations and their implementation.

Lesson Plan No. 3.1.	Course Name: Data Structure using C Topic: Introduction to Linked List	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the Disadvantages of Arrays. b. Articulate the concept and uses of Linked List.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) Ask questions - What do you understand by Structure? - What is the need of structure? - What is structure-pointer concept? - Introduce the drawbacks of arrays. **2. Development** (30 minutes) a. Disadvantages of Arrays -wastage of memory. - Slow Insertion/Deletion Time b. Linked List Basics - Definition - Concept of nodes -Representation c. Uses of Linked list - optimized utilization of space - No advance declaration of memory size - No Empty node d. Why Linked List over array -allocates the memory dynamically -List grows as per the program's demand **3.** Exercise(5 minutes)

	-Ask Students to why link list is preferable over arrays
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3.2	Course Name: Data Structures using C Topic: Types of Linked List	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Identify different types of Linked List. b. Articulate how different types of linked list contribute to the data structures.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions What are the disadvantages of array? What is Linked list? What is node and how it is partitioned? - Introduce the different types of linked list. **2. Development** (30 minutes) a. Classification of linked list -Singly linked lists -Doubly linked lists -Circular linked lists -Circular doubly linked lists b. Concept of different types of Linked List -Singly linked lists (unidirectionallinked list) - Doubly linked lists (bi-directional linked list) - Circular Linked Lists (last node pointing to the head node) -Circular doubly linked lists (mixture of a doubly linked list and a circular linked list) **3. Exercise**(5 minutes) - Ask students regarding the difference in different types of linked list.

	Collect Responses with the help of Nearpod.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings.
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3.3	Course Name: Data Structures using C Topic: Types of Linked List	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Identify different types of Linked List. b. Articulate how different types of linked list contribute to the data structures.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions What are the disadvantages of array? What is Linked list? What is node and how it is partitioned? - Introduce the different types of linked list. **2. Development** (30 minutes) a. Classification of linked list - Singly linked lists - Doubly linked lists - Circular linked lists - Circular doubly linked lists b. Concept of different types of Linked List - Singly linked lists (unidirectional linked list) - Doubly linked lists (bi-directional linked list) - Circular Linked Lists (last node pointing to the head node) - Circular doubly linked lists (mixture of a doubly linked list and a circular linked list) **3. Exercise**(5 minutes) - Ask students regarding the difference in different types of linked list. Collect Responses with the help of Nearpod.

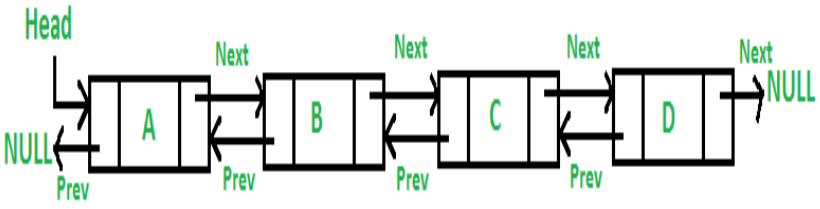
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings.
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No.3.4	Course Name: Data Structures using C Topic: Deletion operations in a linear linked list	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the logic behind the deletion of a node in a linked list. b. Gain knowledge about the types of deletions in a linked list. c. Articulate how nodes and links are updated while deletion process in a linked list.
Teaching Aids (if any)	a. Projector,Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions How do we traverse a linked list? What are the different modes of insertion? How can we count the number of nodes in a single linked list? - Introduce the concept of using deletion of nodes in a linked list. **2. Development** (30 minutes) a. Introduction to deletion process. - to check if list is empty or not. - to find out different modes of deletion in case of a linear linked list b. Deletion in a linear linked list - at the beginning of the linked list - At the end of the linked list - After a specific position c. Code to delete a node linked list - program to give a detailed explanation of deletion of a node in a linked list **3.** Exercise(5 minutes) - Ask students about thedeletionof node at 4th position in a linked list 5->10->15->20->25->30.

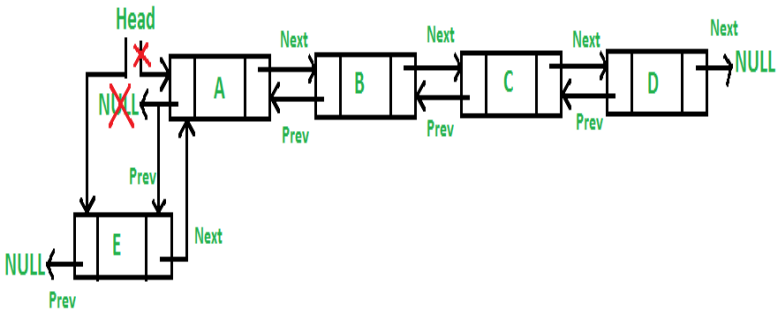
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

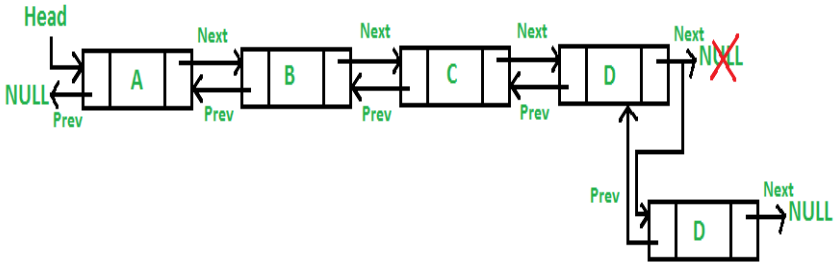
Lesson Plan No.3.5	Course Name: Data Structures using C Topic: Creation and Traversal in a doubly linked list	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the structure a doubly linked list. b. Gain knowledge about the operations that can be performed in a doubly linked list. c. Articulate how nodes are traversed in a doubly linked list.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions What is the structure of a single linked list? What is the difference between single and double linked list? - Discuss the advantages and disadvantages of double linked list over linear linked list **2. Development** (30 minutes) a. Introduce the structure of double linked list - Use of structure to create node b. Declaration of Double Linked Lists - first variable as data - next as a pointer(keep the address of the next node) - prev as a pointer(keep the address of the prev node)  c. Example of a double linked list - program to give a detailed explanation of creation of double linked list. **3. Exercise** (5 minutes)

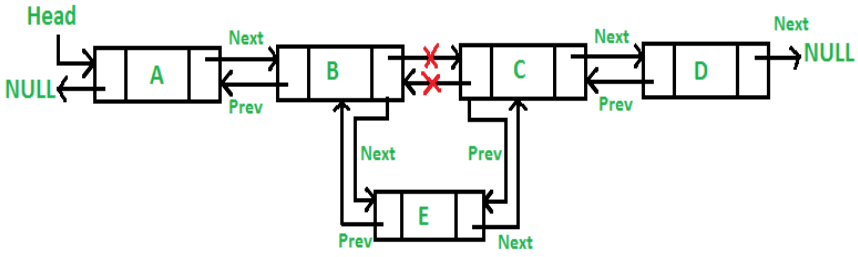
	- Ask students about the nodes and give them a program of traversal of a double linked list.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings.
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No.3.6	Course Name: Data Structure using C Topic: Insertion operations in a doubly linked list	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the types of insertion operations that can be performed in a linked list. b. Gain knowledge about the updation of links of nodes in a linked list c. Articulate how nodes can be inserted at different position in a linked list.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions How can we traverse a doubly linked list? What is the difference between traversal and insertion? - Introduce theconcept of insertion in a doubly linked list. **2. Development** (30 minutes) a. Introduction to insertion - to check if list is empty or not. b. Insertion in a doubly linked list - at the beginning of the linked list  - At the end of the doubly linked list



- After a specific position in a doubly



c. Example to insert a node linked list
- program to give a detailed explanation of insertionin a doubly linked list

3. Exercise(5 minutes)
- Ask students about the insertion and traversalof nodes and give them a program of insertion 30 at 4th position in a linked list 5,10,15,20,25,30.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No.3.7	Course Name: Data Structures using C Topic: Deletion operations in a linked list	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the logic behind the deletion of a node in a doubly linked list. b. Gain knowledge about the types of deletions in a doubly linked list. c. Articulate how nodes and links are updated while deletion process in a doubly linked list.
Teaching Aids (if any)	a. Projector,Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Ask questions How do we traverse a doubly linked list? What are the different modes of insertion? How can we count the number of nodes in a double linked list? - Introduce the concept of using deletion of nodes in a double linked list. **2. Development** (30 minutes) a. Introduction to deletion process. - to check if list is empty or not. - to find out different modes of deletion in case of a double linked list b. Deletion in a double linked list - at the beginning of the linked list - At the end of the linked list - After a specific position c. Code to delete a node linked list - program to give a detailed explanation of deletion of a node in a double linked list **3.** Exercise(5 minutes) - Ask students about thedeletionof node at 4th position in a linked

	list 5->10->15->20->25->30.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Online nptel course on data structures. https://nptel.ac.in/courses/106105085 Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Data Structures Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No.3.8	Course Name: Data Structures using C Topic: Garbage Collection	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of garbage collection in memory management. b. Identify the need for garbage collection in programming languages. c. Explore different garbage collection techniques and their implementations.
Teaching Aids (if any)	a. Projector, Slides b. Code examples demonstrating memory leaks and garbage collection
Teaching Development	**1. Introduction** (5 minutes) - Discuss the concept of dynamic memory allocation. - Ask: What happens to the memory that is no longer in use? - Introduce garbage collection as a mechanism to reclaim unused memory. **2. Development** (30 minutes) a. **Garbage Collection Overview** - Define garbage collection and its role in memory management. - Discuss the problems caused by memory leaks and dangling pointers. b. **Techniques of Garbage Collection** • Explain different garbage collection techniques: - **Reference Counting:** Describe how it works and its limitations (e.g., cyclic references). - **Mark and Sweep:** Explain the marking and sweeping phases. - **Generational Garbage Collection:** Discuss its efficiency in managing different object lifetimes. c. **Implementation in C and Other Languages** - Discuss manual memory management in C and the absence of built-in garbage collection. - Compare with languages like Java or Python, which have

	automatic garbage collection. 3. **Exercise (5 minutes)** - Analyze a piece of code to identify potential memory leaks and discuss how garbage collection could address them.
Closure	Summarize the importance of garbage collection in preventing memory leaks and improving program efficiency. • Suggested Reading • "The Garbage Collection Handbook: The Art of Automatic Memory Management" by Richard Jones, Antony Hosking, and Eliot Moss • Relevant chapters in "Operating Systems Concepts" by Silberschatz, Galvin, and Gagne
Evaluation	•Reflective Questions: Why is garbage collection important? How do different garbage collection techniques work? • Quick quiz on garbage collection concepts and techniques.

Lesson Plan No.3.9	Course Name: Data Structures using C Topic: Header Nodes	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of a header node in linked lists. b. Implement linked lists with header nodes. c. Analyze the advantages of using header nodes in data structures.
Teaching Aids (if any)	a. Projector, Slides b. Code examples demonstrating linked lists with and without header nodes
Teaching Development	**1. Introduction** (5 minutes) - Recap the basic structure of a linked list. - Ask: How can we simplify operations on linked lists, like insertion and deletion at the beginning? - Introduce the concept of a header node and its purpose in linked lists. **2. Development** (30 minutes) a. . **Header Node Overview** - Explain what a header node is and how it differs from regular nodes. - Discuss the advantages of using a header node, such as simplifying list operations and reducing edge cases. b. **Implementation in Linked Lists** - Provide code examples showing how to implement a singly linked list with a header node. - Discuss how insertion, deletion, and traversal operations are affected by the presence of a header node. - Compare the implementation of linked lists with and without header nodes. c. **Applications and Use Cases** - Explore scenarios where header nodes are particularly beneficial, such as in circular linked lists or in algorithms where sentinel nodes are used. **3. Exercise (5 minutes)** -Implement a linked list with a header node and perform basic

	operations like insertion and deletion.
Closure	Summarize the benefits of using header nodes in linked lists and how they simplify operations. Suggested Reading • "Data Structures Using C" by Reema Thareja • Relevant sections in "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein
Evaluation	• Reflective Questions: What are the advantages of using a header node in linked lists? How does it affect the complexity of list operations? • Quick quiz on header nodes and their implementatio

Lesson Plan No.3.10	Course Name: Data Structures using C Topic: Dynamic Memory Management	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of dynamic memory allocation and its importance in programming. b. Implement dynamic memory management using malloc(), calloc(), realloc(), and free() in C. c. Analyze common issues like memory leaks and fragmentation and how to handle them.
Teaching Aids (if any)	a. Projector, Slides b. Code examples demonstrating dynamic memory allocation and deallocation
Teaching Development	**1. Introduction** (5 minutes) - Discuss the limitations of static memory allocation. - Ask: What if the memory required by a program changes during its execution? - Introduce dynamic memory management as a solution to these limitations. **2. Development** (30 minutes) a. **Dynamic Memory Allocation Overview** - Explain the need for dynamic memory allocation in programming. - Introduce the functions malloc(), calloc(), realloc(), and free() in C and their purposes. b. **Implementation in C** - Provide code examples demonstrating the use of malloc() and calloc() for dynamic memory allocation. - Discuss how realloc() is used to resize memory blocks, and the importance of free() in preventing memory leaks. c. **Common Issues and Best Practices** - Discuss common issues like memory leaks, dangling pointers, and memory fragmentation. - Provide best practices for managing dynamic memory, such as checking for NULL after allocation and always freeing allocated memory. **3. Exercise (5 minutes)**

	- Write a program that dynamically allocates memory for an array, resizes it using realloc(), and then frees the memory.
Closure	Recap the importance of dynamic memory management in writing efficient and flexible programs. Suggested Reading • "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie • Online resources on dynamic memory management in C, such as tutorials from GeeksforGeeks or the GNU C Library documentation.
Evaluation	• Reflective Questions: What are the key differences between malloc() and calloc()? How do you prevent memory leaks in dynamic memory management? • Quick quiz on dynamic memory allocation functions and their usage.

Lesson Plan No.4.1	Course Name: Data Structures using C Topic: Binary Trees and Their Representation Using Linked Lists	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of binary trees. b. Describe different types of binary trees and their properties. c. Implement binary trees using linked lists.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Discuss basic tree structures and their importance. - Ask: What is a binary tree? Why is it important? - Introduce the binary tree and its representation using linked lists. **2. Development (30 minutes)** **a. Concept of Binary Trees** - Definition and properties - Types: Full, Complete, Perfect, and Degenerate Trees **b. Representation Using Linked Lists** - Node structure (data, left pointer, right pointer) - Creating a binary tree using linked list nodes **c. Code Implementation** - Provide and explain a sample code for binary tree creation **3. Exercise (5 minutes)** - Create a binary tree with nodes having values 1 to 5.
Closure	- Summarize the lesson and ensure understanding of binary tree representation. - Suggested Reading • "Data Structures and Algorithm Analysis in C" by Mark Allen Weiss

Evaluation	• Reflective Questions: What is a binary tree? How is it represented using linked lists? • Quick quiz on binary tree concepts and representation. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No.4.2	Course Name: Data Structures using C Topic: Operations on Binary Trees	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Perform basic operations on binary trees. b. Implement insertion, deletion, and searching operations. c. Understand how these operations affect the structure of binary trees.
Teaching Aids (if any)	a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction** (5 minutes) - Discuss common operations on binary trees. - Ask: What operations do you think are crucial for binary trees? **2. Development (30 minutes)** a. **Insertion Operation** - Insert nodes into binary trees - Explain insertion logic with examples b. **Deletion Operation** - Remove nodes and restructure the tree - Discuss different deletion scenarios c. **Searching Operation** - Search for nodes in a binary tree - Explain search algorithms and complexity **3. Exercise (5 minutes)** • Implement insertion of nodes with values 6 to 10 in an existing binary tree.
Closure	- Recap the operations discussed and their implementations. - Suggested Reading • "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and

	Stein
Evaluation	• Reflective Questions: How does insertion affect the binary tree? What happens during deletion? • Quick quiz on binary tree operations.

Lesson Plan No.4.3	Course Name: Data Structures using C Topic: Traversal Algorithms for Binary Trees	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand different binary tree traversal methods. b. Implement pre-order, in-order, and post-order traversals c. Analyze the efficiency of traversal algorithms.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for traversal
Teaching Development	**1. Introduction** (5 minutes) a. Explain the need for tree traversal. b. Ask: Why do we need different traversal methods? **2. Development (30 minutes)** a. Explain the need for tree traversal. b. Ask: Why do we need different traversal methods? c. **Traversal Methods** - Pre-order traversal - In-order traversal - Post-order traversal d. **Code Implementation** - Provide code for each traversal method - Explain code and output examples c. **Efficiency Analysis** - Discuss time complexity of different traversals **3. Exercise (5 minutes)** - Implement and test all three traversal methods on a binary tree.
Closure	Summarize traversal methods and their use cases. • Suggested Reading "Algorithms" by Robert Sedgewick and Kevin Wayne
Evaluation	• Reflective Questions: What are the differences between traversal methods? How does traversal affect tree structure? • Quick quiz on traversal algorithms.

Lesson Plan No.4.4	Course Name: Data Structures using C Topic: Applications of Binary Trees	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Identify real-world applications of binary trees b. Understand how binary trees are used in practical scenarios. c. Implement binary tree-based solutions to problems.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for traversal
Teaching Development	**1. Introduction** (5 minutes) - Discuss the relevance of binary trees in various applications. - Ask: Can you think of any real-world applications of binary trees? **2. Development (30 minutes)** a. **Applications Overview** - Expression trees - Huffman coding trees - Binary search trees for databases b. **Case Studies** - Analyze examples of binary tree applications - Discuss advantages and limitations **3. Exercise (5 minutes)** - Solve a problem using a binary tree, such as constructing an expression tree.
Closure	Recap the applications and discuss their impact. • Suggested Reading "Computer Algorithms: Introduction to Design and Analysis" by Sara Baase and Alan Kleinberg
Evaluation	• Reflective Questions: What are the key applications of binary trees? How do they benefit specific use cases? • Quick quiz on applications of binary trees.

Lesson Plan No.4.5	Course Name: Data Structures using C Topic: Threaded Binary Trees and Their Traversal Algorithms	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of threaded binary trees. b. Implement threading in binary trees. c. Perform traversal operations on threaded binary trees.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for traversal
Teaching Development	**1. Introduction (5 minutes)** - Explain the concept of threading in binary trees. - Ask: What challenges in tree traversal can be addressed by threading? **2. Development (30 minutes)** a. **Concept of Threaded Binary Trees** - Definition and types (single-threaded, double-threaded) - Benefits of threading b. **Traversal Algorithms** - Implement in-order traversal in threaded trees - Discuss other traversal methods and their efficiency **3. Exercise (5 minutes)** - Implement threading in a binary tree and perform in-order traversal.
Closure	• Summarize threaded binary trees and their advantages. • Suggested Reading "Data Structures and Algorithms in C++" by Adam Drozdek
Evaluation	• Reflective Questions: How does threading improve tree traversal? What are the different types of threading? • Quick quiz on threaded binary trees and traversal.

Lesson Plan No.4.6	Course Name: Data Structures using C Topic: Heterogeneous Binary Trees	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of heterogeneous binary trees. b. Implement a heterogeneous binary tree structure. c. Analyze the use cases and limitations of heterogeneous binary trees.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for traversal
Teaching Development	**1. Introduction (5 minutes)** - Define heterogeneous binary trees and their differences from homogeneous trees. - Ask: What scenarios require heterogeneous tree structures? **2. Development (30 minutes)** a. **Concept and Implementation** - Definition and structure - Example implementations in code b. **Use Cases and Limitations** - Analyze where heterogeneous trees are useful - Discuss potential issues **3. Exercise (5 minutes)** - Implement a simple heterogeneous binary tree and explore its structure.
Closure	• Recap the concept and applications of heterogeneous binary trees. • Suggested Reading • "Data Structures and Algorithms" by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft
Evaluation	• Reflective Questions: What makes a binary tree heterogeneous? How do you implement such a tree? • Quick quiz on heterogeneous binary trees.

Lesson Plan No.4.7	Course Name: Data Structures using C Topic: Graphs and Their Representations	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand different ways to represent graphs. b. Implement adjacency matrix and adjacency list representations. c. Analyze the pros and cons of different representations.
Teaching Aids (if any)	a. Projector, Slides b. Visual aids for graph representations
Teaching Development	**1. Introduction (5 minutes)** - Discuss what graphs are and their importance. - Ask: How do you think we can represent a graph in memory? **2. Development (30 minutes)** a. **Adjacency Matrix Representation** - Explain and provide examples - Discuss space and time complexity b. **Adjacency List Representation** - Explain and provide examples - Compare with adjacency matrix c. **Other Representations** - Discuss alternative methods like edge list • **Exercise (5 minutes)** • Implement a graph using both adjacency matrix and adjacency list.
Closure	•Recap graph representations and their importance in various algorithms. • Suggested Reading • "Graph Theory with Applications to Engineering and Computer

	Science" by Narsingh Deo
Evaluation	• Reflective Questions: What are the advantages of using an adjacency list over an adjacency matrix? • Quick quiz on graph representation methods.

Lesson Plan No.4.8	Course Name: Data Structures using C Topic: Traversal Methods for Graphs	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the different methods to traverse graphs. b. Implement Depth First Search (DFS) and Breadth First Search (BFS) c. Analyze the efficiency and use cases of each traversal method.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for DFS and BFS
Teaching Development	**1. Introduction (5 minutes)** - Explain the need for graph traversal. - Ask: Why do we need different traversal methods in graphs? **2. Development (30 minutes)** a. **Depth First Search (DFS)** - Explain the algorithm and its working - Provide a code example and discuss complexity b. **Breadth First Search (BFS)** - Explain the algorithm and its working - Provide a code example and discuss complexity c. **Use Cases and Comparisons** - Analyze when to use DFS vs BFS **3. Exercise (5 minutes)** - Implement both DFS and BFS on a sample graph.
Closure	• Summarize traversal methods and their applications in graph algorithms. • Suggested Reading • "Introduction to Graph Theory" by Robin J. Wilson
Evaluation	• Reflective Questions: How does DFS differ from BFS? When would you use one over the other? • Quick quiz on DFS and BFS.

Lesson Plan No.4.9	Course Name: Data Structures using C Topic: Optimum Search Trees	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of optimum search trees. b. Implement and analyze the construction of optimal binary search trees. c. Explore the use cases where optimum search trees provide efficiency.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for constructing optimum search trees
Teaching Development	**1. Introduction (5 minutes)** - Discuss the importance of search efficiency in data structures. - Ask: What factors can affect the efficiency of search operations? - Introduce the concept of optimum search trees and their purpose. **2. Development (30 minutes)** a. **Optimum Search Trees Overview** - Definition and significance - Difference between general and optimum search trees b. **Construction of Optimal Binary Search Trees** - Dynamic programming approach - Explain cost functions and tree structure - Provide a step-by-step example of constructing an optimal binary search tree c. **Use Cases and Efficiency** - Explore scenarios where optimum search trees are used - Discuss the time complexity and efficiency of search operations **3. Exercise (5 minutes)** - Construct an optimal binary search tree for a set of given probabilities and keys.

Closure	• Summarize the lesson by discussing the benefits of optimum search trees in improving search efficiency. • Suggested Reading • "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein (Chapter on Dynamic Programming and Optimum Search Trees)
Evaluation	• Reflective Questions: What is the main advantage of using an optimum search tree? How is it constructed? • Quick quiz on the concepts and construction of optimum search trees.

Lesson Plan No.4.10	Course Name: Data Structures using C Topic: AVL Trees	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of AVL trees and their properties. b. Implement AVL tree insertion and rotation operations. c. Analyze the balance factor and the need for balancing in binary search trees.
Teaching Aids (if any)	a. Projector, Slides b. Code examples for AVL tree operations
Teaching Development	**1. Introduction (5 minutes)** - Introduce the concept of self-balancing trees. - Ask: What challenges do we face when a binary search tree becomes unbalanced? - Explain the need for AVL trees to maintain balance in binary search trees. **2. Development (30 minutes)** a. **AVL Trees Overview** - Definition and properties - Balance factor and height of AVL trees b. **Insertion and Rotations** - Explain the process of insertion in an AVL tree - Discuss different types of rotations (single and double) to maintain balance - Provide code examples for insertion and rotation operations c. **Balancing and Efficiency** - Analyze the importance of balancing in AVL trees - Discuss time complexity for search, insertion, and deletion operations in AVL trees **3. Exercise (5 minutes)** - Insert a sequence of values into an AVL tree and demonstrate the necessary rotations.

Closure	• Summarize the lesson by discussing the significance of AVL trees in maintaining balanced binary search trees. • Suggested Reading • "Data Structures and Algorithm Analysis in C" by Mark Allen Weiss (Chapter on AVL Trees)
Evaluation	• Reflective Questions: What is the purpose of balancing in AVL trees? How do rotations help maintain balance? • Quick quiz on AVL tree properties and operations.

Lesson Plan No. 5.1	Course Name: Data Structure Topic: Bubble Sort and Quick Sort	Course No.: COM-201

Objectives	At the end of the lesson, students shall be able to: a. Understand and implement Exchange Sort algorithms. b. Differentiate between Bubble Sort and Quick Sort. c. Evaluate the efficiency and performance of Bubble Sort and Quick Sort.
Teaching Aids (if any)	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction (5 minutes)** - Ask questions: - What are sorting algorithms? - How do Exchange Sort algorithms differ from other sorting methods? - Discuss the concept of Exchange Sort and its importance. 2. **Development (30 minutes)** a. Bubble Sort - Explain the algorithm and its process - Demonstrate Bubble Sort with a step-by-step example - Write a program to implement Bubble Sort b. Quick Sort - Explain the algorithm and its process - Discuss the concept of pivot and partitioning - Demonstrate Quick Sort with an example - Write a program to implement Quick Sort c. Performance Comparison - Compare Bubble Sort and Quick Sort in terms of time complexity and performance - Exercise (5 minutes) - Ask students to write programs for both Bubble Sort and Quick Sort and analyze their performance.
Closure	Closure: 1. Summarize the Lesson Learning Outcomes and seek student affirmation. 2. Suggested Reading: - Online NPTEL course on data structures: NPTEL Course Spend 5 minutes consolidating the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Bubble sort and quick sort Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5.2	Course Name: Data Structure Topic: Selection Sort and its Comparison with Exchange Sort Algorithms	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand and implement Selection Sort. b. Compare Selection Sort with Exchange Sort algorithms. c. Discuss the applications and limitations of Selection Sort.
Teaching Aids (if any)	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
Teaching Development	**1. Introduction (5 minutes)** Ask questions: - What are sorting algorithms? - How do Exchange Sort algorithms differ from other sorting methods? - Discuss the concept of Exchange Sort and its importance. 2. **Development (30 minutes)** a. Bubble Sort - Explain the algorithm and its process - Demonstrate Bubble Sort with a step-by-step example - Write a program to implement Bubble Sort b. Quick Sort - Explain the algorithm and its process - Discuss the concept of pivot and partitioning - Demonstrate Quick Sort with an example - Write a program to implement Quick Sort c. Performance Comparison - Compare Bubble Sort and Quick Sort in terms of time complexity and performance **3. Exercise (5 minutes)** - Ask students to write programs for both Bubble Sort and Quick Sort and analyse their performance.
Closure	1. Summarize the Lesson Learning Outcomes and seek student affirmation. 2. Suggested Reading: - Online NPTEL course on data structures: https://nptel.ac.in/courses/106105085 - Spend 5 minutes consolidating the learning. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	2. Nearpod Quiz on Bubble Sort and quick sort. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5.3	Course Name: Data Structure using C Topic: Selection Sort and its Comparison with Exchange Sort Algorithms	Course No.: COM-201

Objectives	At the end of the lesson, students shall be able to: a. Understand and implement Selection Sort. b. Compare Selection Sort with Exchange Sort algorithms. c. Discuss the applications and limitations of Selection Sort.
Teaching Aids (if any)	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction (5 minutes)** - Ask questions: - What is Selection Sort? - How does it compare to Bubble Sort and Quick Sort? - Discuss the Selection Sort algorithm and its use cases. 2. **Development (30 minutes)** a. Selection Sort - Explain the Selection Sort algorithm and its process - Demonstrate Selection Sort with a step-by-step example - Write a program to implement Selection Sort b. Comparison with Exchange Sort - Compare Selection Sort with Bubble Sort and Quick Sort in terms of time complexity and efficiency c. Applications and Limitations - Discuss scenarios where Selection Sort is beneficial and its limitations **3. Exercise (5 minutes)** • Ask students to implement Selection Sort and compare its performance with Bubble Sort and Quick Sort Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and seek student affirmation. 2. Suggested Reading: - Online NPTEL course on data structures: https://nptel.ac.in/courses/106105085 - Spend 5 minutes consolidating the learnings. Spend 5 minutes to wrap up and consolidate the learnings

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Selection sort Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5.4	Course Name: Data Structure using C Topic: Insertion Sort and its Comparison with Other Sorting Methods	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand and implement Insertion Sort. b. Compare Insertion Sort with other sorting methods. c. Analyze the performance and use cases of Insertion Sort.
Teaching Aids (if any)	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction (5 minutes)** - Ask questions: - What is Insertion Sort? - How does it differ from Bubble Sort and Quick Sort? - Discuss the concept of Insertion Sort and its typical use cases. 2. **Development (30 minutes)** a. Insertion Sort - Explain the Insertion Sort algorithm and its process - Demonstrate Insertion Sort with a step-by-step example - Write a program to implement Insertion Sort b. Comparison with Other Sorting Methods - Compare Insertion Sort with Bubble Sort, Quick Sort, and Selection Sort in terms of performance and efficiency c. Performance Analysis - Discuss the time complexity and practical applications of Insertion Sort 3. Exercise (5 minutes) • Ask students to implement Insertion Sort and compare its performance with other sorting algorithms. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and seek student affirmation. 2. Suggested Reading: • Online NPTEL course on data structures: https://nptel.ac.in/courses/106105085 • Spend 5 minutes consolidating the learnings.

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on insertion sort Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5.5	Course Name: Data Structure Using C Topic: Merge Sort and its Comparison with Other Sorting Algorithms	Course No.: COM-201

Objectives	At the end of the lesson the student shall be able to: a. Understand and implement Merge Sort. b. Compare Merge Sort with other sorting algorithms such as Quick Sort and Bubble Sort c. Discuss the advantages and applications of Merge Sort.
Teaching Aids (if any)	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction (5 minutes)** - Ask questions: o What is Merge Sort? o How does Merge Sort work compared to other sorting algorithms? - Discuss the concept and benefits of Merge Sort. 2. **Development (30 minutes)** a. Merge Sort - Explain the Merge Sort algorithm and its divide-and-conquer approach - Demonstrate Merge Sort with a step-by-step example - Write a program to implement Merge Sort b. Comparison with Other Sorting Methods - Compare Merge Sort with Quick Sort and Bubble Sort in terms of time complexity and performance c. Advantages and Applications - Discuss the advantages of Merge Sort, including its stability and efficiency in handling large data sets 3. **Exercise (5 minutes)** - Ask students to implement Merge Sort and analyze its performance compared to Quick Sort and Bubble Sort.
Closure	1. Summarize the Lesson Learning Outcomes and seek student affirmation. 2. Suggested Reading: - Online NPTEL course on data structures: https://nptel.ac.in/courses/106105085 - Spend 5 minutes consolidating the learnings. Spend 5 minutes to wrap up and consolidate the learnings

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Merge sort Spend 5 minutes to evaluate student assimilation of the lesson contents