



Kot Bhalwal, Jammu



Model Institute of Engineering  
& Technology (Autonomous)  
Dr. Arun K. Gupta Teaching-Learning Centre

## Department of CSE

### Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Data Structures Using C
2.	Course Code	COM-201
3.	Academic Year	2024-25
4.	Semester	2 <sup>nd</sup>
5.	Number of Lesson plans	45
6.	Faculty Assigned	Anurag Slathia Adarsh Kumar Singh

Faculty Signature



Version 1.1



Please Do Not Print Unless Necessary



Kot Bhalwal, Jammu

Model Institute of Engineering  
& Technology (Autonomous)  
Dr. Arun K. Gupta Teaching-Learning Centre

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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Use of Near pod tool for online quiz
<b>Teaching Development</b>	<ol style="list-style-type: none"><li>1. <b>Introduction</b> (5 minutes)<ul style="list-style-type: none"><li>- Ask questions What do you understand by number system?  What type of number system do you use for general purpose?</li><li>- Introduce the concept of number system</li><li>- Talk about the need of different number systems</li></ul></li> <li>2. <b>Development</b> (30 minutes)<ol style="list-style-type: none"><li>a. Concept of different number systems<ul style="list-style-type: none"><li>- Introduce different number systems</li><li>- Talk about Decimal, Binary, Octal, HexaDecimal</li></ul></li> <li>b. Conversion of different number systems<ul style="list-style-type: none"><li>- Conversion of decimal to binary and binary to decimal.</li><li>- Conversion of decimal to octal, octal to decimal</li><li>- Conversion of octal to binary and binary to octal</li><li>- Conversion of decimal to hexadecimal and hexadecimal to decimal.</li><li>- Conversion of hexadecimal to binary and binary to hexadecimal.</li></ul></li></ol></li></ol>



<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on number system.  <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li><li>3. Homework<ul style="list-style-type: none"><li>- Convert the given numbers to another forms and submit the answers on Google classroom</li></ul></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learning</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Use of Nearpod tool for online quiz
<b>Teaching Development</b>	<ol style="list-style-type: none"><li>1. <b>Introduction</b> (5 minutes)<ul style="list-style-type: none"><li>- Ask questions</li><li>- What do you understand by the word data?</li><li>- What are the different types of data present in C language?</li><li>- Do you know why is it important to store data in a particular manner?</li><li>- Introduce the definition of Data Structures.</li><li>- Talk about thereal world examples of data structures.</li><li>- Highlight the important characteristics of data structures.</li></ul></li><li>2. <b>Development</b> (30 minutes)<ol style="list-style-type: none"><li>a. Data Structure basics<ul style="list-style-type: none"><li>- Introduce the concept of data structure including its commonly used applications like plotting of graphs, text editing etc.</li></ul></li><li>b. Need of data structures<ul style="list-style-type: none"><li>- Processor speed</li><li>- Data Search</li><li>- Multiple request</li></ul></li><li>c. Classification of data structures<ul style="list-style-type: none"><li>- Introducing primitive data structures(built-in data types) and non-primitive data structures(derived data types)</li><li>-Show flow chart includingexamples of primitive data structure (int,char,floatetc) and non primitive data structure(array, linked list, stack,queue , graphs,trees).</li><li>- Give overview of non-primitive data structure.</li></ul></li><li>d. Need of data structures<ul style="list-style-type: none"><li>- Processor speed</li><li>- Data Search</li><li>- Multiple request</li></ul></li></ol></li></ol>



	<p>e. Basic Operations performed on Data Structure</p> <ul style="list-style-type: none"><li>- Traversing</li><li>- Insertion</li><li>- Deletion</li><li>- Sorting</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li><li>3. Homework<ul style="list-style-type: none"><li>- Create a presentation highlighting data structure concepts and submit on Google classroom</li></ul></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Near pod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Articulate the fundamental concept of Arrays.</li> <li>Understand the need of Arrays.</li> <li>Identify different types of Arrays.</li> <li>Operations performed on Array.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector ,Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>Introduction</b> (5 minutes)           <ul style="list-style-type: none"> <li>Ask questions               <ul style="list-style-type: none"> <li>What do you understand by Array?</li> <li>What is the use of Array?</li> <li>How does the array store Data?</li> </ul> </li> <li>Introduce the definition of Array.</li> <li>Talk about the examples of Array.</li> <li>Highlight the different types of Array.</li> </ul> </li> <li><b>Development</b> (30 minutes)           <ol style="list-style-type: none"> <li>Array basics               <ul style="list-style-type: none"> <li>Declaration of Array</li> <li>Initializing Array</li> <li>Accessing elements of Array</li> </ul> </li> <li>Need of Arrays               <ul style="list-style-type: none"> <li>Stores multiple number of elements.</li> <li>Stores any kind of primitive data type.</li> </ul> </li> <li>Classification of Arrays               <ul style="list-style-type: none"> <li>1-D Array</li> <li>Multi-Dimensional Array</li> </ul> </li> <li>Basic Operations performed on Arrays               <ul style="list-style-type: none"> <li>Traversing</li> <li>Insertion</li> <li>Deletion</li> <li>Sorting</li> <li>Searching</li> </ul> </li> </ol> </li> </ol>



	<p>e. Concept of Searching in Arrays</p> <ul style="list-style-type: none"><li>- 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).</li><li>- Binary Search(Searching a sorted array by repeatedly dividing the search interval in half.)</li></ul> <p>3. Exercise(5 minutes)</p> <ul style="list-style-type: none"><li>- Ask Students to write down the program for Searching elements in an array.</li><li>- Asking them which technique is better.</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the different sorting techniques.</li> <li>Articulate basic concepts of Multi-Dimensional Arrays.</li> <li>Perform manipulations on Arrays</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask questions What are the different operations that can be performed on array?</li> <li>What is Linear Search?</li> <li>What is Binary Search?</li> <li>- Introduce the concept of sorting.</li> <li>- Highlight the different types of Sorting techniques.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ol style="list-style-type: none"> <li>Sorting of array (ascending/descending order)</li> <li>Different sorting techniques           <ul style="list-style-type: none"> <li>-Bubble Sort</li> <li>-Selection Sort</li> <li>-Insertion Sort</li> <li>-Merge Sort</li> <li>-Quicksort</li> <li>-Radix Sort</li> <li>- Shell sort</li> </ul> </li> <li>Introduction to multi-dimensional array           <ul style="list-style-type: none"> <li>- Declaration</li> <li>- Initialization</li> <li>- Accessing elements in 2-D array</li> </ul> </li> <li>Manipulations on 2-D array           <ul style="list-style-type: none"> <li>- Addition</li> </ul> </li> </ol>



	<ul style="list-style-type: none"><li>- Subtraction</li><li>- Multiplication</li></ul> <p>3. Exercise(5 minutes)</p> <ul style="list-style-type: none"><li>- Write down a program to add rows of 2-D array</li><li>- Write down a program to add columns of 2-D array</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>a. Articulate the fundamental concept of Pointers.</li> <li>b. Understand Pointers and Array.</li> <li>c. Advantages of pointer.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>a. Projector ,Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask questions How to initialize 2-D array? What is pointer?</li> <li>- Introduce the definition of pointer.</li> <li>- Give an example of pointer.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>a. Pointer basics               <ul style="list-style-type: none"> <li>- Discussion of pointer with example.</li> <li>- Declaration of pointer.</li> <li>- A detailed example of pointers</li> </ul> </li> <li>b. Pointers and array               <ul style="list-style-type: none"> <li>- Relation of arrays and pointers.</li> </ul> </li> <li>c. Advantages of pointer               <ul style="list-style-type: none"> <li>- Reduces the code and improves the performance.</li> <li>- Return multiple values from a function.</li> <li>- access any memory location</li> </ul> </li> <li>d. Uses of Pointers               <ul style="list-style-type: none"> <li>-Dynamic memory allocation</li> <li>- Arrays, Functions, and Structures</li> </ul> </li> </ol> </li> <li><b>3. Exercise(5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask Students to write down the program to find the sum of n</li> </ul> </li> </ol>



	numbers with arrays and pointers.
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> Spend 5 minutes to wrap up and consolidate the learnings.</li></ol>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of De-referencing.</li> <li>b. Know the need of De-referencing</li> <li>c. Articulate the basics of Void pointers.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask questions What do you understand by pointer? What is the output of the program (from pointers and array) displayed on the screen?</li> <li>- Introduce the concept of de-referencing.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. De- Referencing basics <ul style="list-style-type: none"> <li>- Representation of de-referencing pointer.</li> <li>- Steps to de-reference a pointer.</li> <li>- Detailed Example</li> </ul> </li> <li>b. Need of dereferencing a pointer <ul style="list-style-type: none"> <li>- to access or manipulate the data stored at the memory location, which is pointed by the pointer.</li> <li>- operation applied will directly affect the value of the variable that it points to.</li> </ul> </li> <li>c. Basics of Void Pointers <ul style="list-style-type: none"> <li>- Definition</li> <li>- Example of Void pointer</li> </ul> </li> <li>d. Advantages of Void Pointer <ul style="list-style-type: none"> <li>- Implement generic functions in C</li> <li>- malloc() and calloc() return void * type and this can help allocate memory of any data type</li> </ul> </li> </ul>



	<p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"><li>- Ask Students to write down the program giving example of void pointer.</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of Dynamic memory allocation.</li> <li>b. Identify and understand the basic functions used for dynamic memory allocation.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask questions What is De-Referencing? What is the function of void pointer?</li> <li>- Introduce the need of Dynamic memory allocation.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>a. Dynamic memory allocation basics <ul style="list-style-type: none"> <li>- definition of Dynamic memory allocation</li> </ul> </li> <li>b. Library functions for Dynamic memory allocation <ul style="list-style-type: none"> <li>- malloc(memory allocation)</li> <li>- calloc(contiguous allocation)</li> <li>- free(de-allocate)</li> <li>- realloc(re-allocation)</li> </ul> </li> <li>c. Examples of all the functions <ul style="list-style-type: none"> <li>- Difference between all the library functions (malloc, calloc, free, realloc) with the help of examples.</li> </ul> </li> </ol> </li> <li><b>3. Exercise(5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask Students to identify the use of each function required in dynamic memory allocation.</li> <li>- Collect the responses using Nearpod.</li> </ul> </li> </ol>



<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>a. Articulate the concept of Structures.</li> <li>b. Understanding pointer to structures.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz.</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- 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()</li> <li>- Introduce the basic concept of data structures</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>a. Structure basics               <ul style="list-style-type: none"> <li>- Defining a Structure</li> <li>- Accessing Structure Member</li> <li>- A well defined example</li> </ul> </li> <li>b. Structure to pointer               <ul style="list-style-type: none"> <li>- Definition</li> <li>- Example of structure pointer.</li> </ul> </li> </ol> </li> <li><b>3. Exercise(5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask Students to determine output of the program shown on the screen.</li> <li>- Collect the responses using Nearpod.</li> </ul> </li> </ol>
<b>Closure</b>	<ol style="list-style-type: none"> <li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li> <li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li> </ol>



	Spend 5 minutes to wrap up and consolidate the learnings
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>a. Understand the stack data structure</li> <li>b. Gain knowledge about different types of basic operations that can be performed in a stack.</li> <li>c. Articulate the applications of stack.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <p>Ask questions</p> <ul style="list-style-type: none"> <li>- What is the difference between arrays and linked lists?</li> <li>- What is the importance of both?</li> <li>- What is the difference between physical and logical data structures?</li> <li>- Introduce the concept of stacks</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>a. Introduction to stacks.               <ul style="list-style-type: none"> <li>- Examples of stacks in daily life.</li> <li>- Significance of top of the stack.</li> </ul> </li> <li>b. Basic operations on stacks               <ul style="list-style-type: none"> <li>- push</li> <li>- pop</li> <li>- peek</li> <li>- isEmpty</li> <li>- isFull</li> </ul> </li> <li>c. Applications of stacks               <ul style="list-style-type: none"> <li>- String reversal</li> <li>- UNDO/REDO</li> <li>- Recursion</li> <li>- DFS(Depth First Search)</li> <li>- Backtracking</li> <li>- Expression conversion</li> <li>- Memory management</li> </ul> </li> </ol> </li> </ol>



	<p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"><li>- Ask students about the insertion and deletion onto a stack</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the different types of implementations of stacks.</li> <li>b. Gain knowledge about implementation of stacks in C using arrays</li> <li>c. Articulate about hoe to perform different stack operations using arrays.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b> Ask questions</p> <ul style="list-style-type: none"> <li>- What is a stack data structure?</li> <li>- What is the importance static and dynamic stack implementation?</li> <li>- Introduce theconcept of implementation of stacks.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. Introduction to arrays and stacks. <ul style="list-style-type: none"> <li>- Significance of using arrays for implementing stack.</li> <li>- Explaining the array implementation of stack.</li> </ul> </li> <li>b. Implementation of basic operations on stacks using arrays <ul style="list-style-type: none"> <li>- push</li> <li>- pop</li> <li>- peek</li> <li>- isEmpty</li> <li>- isFull</li> </ul> </li> </ul> <p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask students about theinsertion and deletion onto a stack</li> <li>- 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?</li> </ul> <p>b,a,c</p> <p>b,c,a</p>



	<p>c,a,b</p> <p>a,b,c</p>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the different types of notations for expressions.</li> <li>Gain knowledge about parenthesis checking.</li> <li>Articulate about the conversion of infix to postfix expression.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>Ask questions               <ul style="list-style-type: none"> <li>What are the applications of stacks?</li> <li>Why do we need to check the balancing of parenthesis?</li> <li>Why do we need to convert infix into postfix expression?</li> <li>Introduce the concept of infix, prefix and postfix expressions.</li> </ul> </li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>Introduction to types of expressions with examples.               <ul style="list-style-type: none"> <li>Infix</li> <li>Postfix</li> <li>Prefix</li> </ul> </li> <li>Briefing the rules that are to be followed for the conversion of infix expression to the postfix expression. Explaining with examples.</li> </ol> </li> <li><b>3. Exercise(5 minutes)</b> <ul style="list-style-type: none"> <li>Ask students to evaluate the postfix expression for the following               <math display="block">A+b*c</math> <math display="block">A-b/c*d+e</math> <math display="block">K + L - M*N + (O^P) * W/U/V * T + Q</math> <math display="block">A-B+(M^N)*(O+P)-Q/R^S*T+Z</math> </li> </ul> </li> </ol>
<b>Closure</b>	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.



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



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the concept of postfix evaluation.</li> <li>Gain knowledge about the infix to prefix conversion.</li> <li>Articulate about how to evaluate an expression given in postfix notation.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <p>Ask questions</p> <ul style="list-style-type: none"> <li>What is the difference between postfix and prefix notation?</li> <li>Why do we need to perform the conversions?</li> <li>Introduce the concept of postfix expression evaluation and infix-prefix conversion.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>Introduction to postfix evaluation               <ul style="list-style-type: none"> <li>Rules to evaluate a postfix expression.</li> <li>Examples for evaluation                    456*+                    53+83-*                    35*62/+4-                    2 3 1 * + 9 -                    53+62/*35*+                 </li> </ul> </li> <li>Infix to prefix conversion               <ul style="list-style-type: none"> <li>discuss the rules for infix to prefix conversion.</li> </ul> </li> </ol> </li> <li><b>3. Exercise(5 minutes)</b> <ul style="list-style-type: none"> <li>Ask studentsto convert the given infix expression to prefix                (A+B)+C-(D-E)^F             </li> </ul> </li> </ol>
<b>Closure</b>	<ol style="list-style-type: none"> <li>Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li> <li>Suggested Reading            Online nptel course on data structures.            <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> </li> </ol>



	Spend 5 minutes to wrap up and consolidate the learnings
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the queue data structure</li> <li>Gain knowledge about different types of basic operations that can be performed in a queue.</li> <li>Articulate the applications of queue.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b> Ask questions</p> <ul style="list-style-type: none"> <li>What is the principle of working of a stack?</li> <li>What is the importance of top variable?</li> <li>What is the difference between insertion and deletion in terms of top variable?</li> <li>Introduce the concept of queues.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ol style="list-style-type: none"> <li>Introduction to queues.           <ul style="list-style-type: none"> <li>Examples of queues in daily life.</li> <li>Significance of maintaining two variables in case of queue.               <ul style="list-style-type: none"> <li>Front</li> <li>Rear</li> </ul> </li> </ul> </li> </ol> <div data-bbox="555 1473 1248 1886" data-label="Diagram"> </div> <ol style="list-style-type: none"> <li>Basic operations on stacks           <ul style="list-style-type: none"> <li>enqueue</li> <li>dequeue</li> <li>peek</li> </ul> </li> </ol>



	<ul style="list-style-type: none"><li>- display</li><li>- isEmpty</li><li>- isFull</li></ul> <p>c. Applications of stacks</p> <ul style="list-style-type: none"><li>- Managing requests on a single shared resource</li><li>- Handling hardware or real-time systems interrupts</li><li>- Handling website traffic</li><li>- Routers and switches in networking</li><li>- Maintaining the playlist in media players</li></ul> <p>3. Exercise(5 minutes)</p> <ul style="list-style-type: none"><li>- Ask students about the insertion and deletion in a queue</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> <a href="https://www.geeksforgeeks.org/queue-data-structure/">https://www.geeksforgeeks.org/queue-data-structure/</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the pop operation in the context of a stack data structure.</li> <li>Implement the pop operation using arrays and linked lists.</li> <li>Analyze the edge cases and potential errors in the pop operation.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Code examples for pop operation in stacks</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>Recap the concept of stacks and their LIFO (Last In, First Out) principle.</li> <li>Ask: What happens when we try to remove an element from a stack?</li> <li>Introduce the pop operation as the mechanism for removing elements from a stack.</li> </ul> </li> <li><b>Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>Pop Operation Overview</b> <ul style="list-style-type: none"> <li>Explain the logic behind the pop operation.</li> <li>Discuss the steps involved: checking for an empty stack, retrieving the top element, and updating the stack pointer.</li> </ul> </li> <li><b>Implementation Using Arrays</b> <ul style="list-style-type: none"> <li>Provide a code example of implementing the pop operation using an array-based stack.</li> <li>Discuss potential issues such as stack underflow.</li> </ul> </li> <li><b>Implementation Using Linked Lists</b> <ul style="list-style-type: none"> <li>Provide a code example of implementing the pop operation using a linked list-based stack.</li> <li>Compare the array and linked list implementations.</li> </ul> </li> </ol> </li> <li><b>Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>Write code to implement the pop operation in a stack and handle</li> </ul> </li> </ol>



	edge cases. -
<b>Closure</b>	Summarize the importance of the pop operation and how it is implemented in different stack structures. <ul style="list-style-type: none"><li>• Suggested Reading  "Data Structures and Algorithms in C" by Adam Drozdek</li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What are the key steps in the pop operation? How does the implementation differ between arrays and linked lists?</li><li>• Quick quiz on stack operations, focusing on pop.</li></ul>



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Code examples for push operation in stacks
<b>Teaching Development</b>	<b>1. Introduction (5 minutes)</b>  - 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.  <b>2. Development (30 minutes)</b>  <b>a. Push Operation Overview</b>  - Explain the logic of the push operation. - Discuss the process: checking for stack overflow, adding the element, and updating the stack pointer.  <b>b. Implementation Using Arrays</b>  - Provide a code example of implementing the push operation using an array-based stack. - Discuss handling of stack overflow.  <b>c. Implementation Using Linked Lists</b>  - 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.  <b>3. Exercise (5 minutes)</b>  - Write code to implement the push operation and handle edge



	cases like stack overflow.
<b>Closure</b>	Summarize the push operation and its role in stack manipulation. <ul style="list-style-type: none"><li>• Suggested Reading  "Data Structures Using C" by Reema Thareja</li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What are the potential issues when implementing the push operation? How can these be handled?</li><li>• Quick quiz on stack operations, focusing on push.</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>a. Understand the concept of multiple queues and their applications.</li> <li>b. Implement multiple queues using array and linked list data structures.</li> <li>c. Analyze the scenarios where multiple queues are beneficial.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Visual aids to demonstrate multiple queues</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Discuss the basic concept of a queue.</li> <li>- Ask: How would multiple queues be useful in a system?</li> <li>- Introduce the concept of multiple queues and their applications in operating systems, scheduling, and more.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>a. Multiple Queues Overview</b> <ul style="list-style-type: none"> <li>- Explain the structure and use cases of multiple queues.</li> </ul> </li> <li><b>b. Implementation Using Arrays</b> <ul style="list-style-type: none"> <li>- Provide code examples for implementing multiple queues using arrays.</li> <li>- Discuss the challenges and limitations.</li> </ul> </li> <li><b>c. Implementation Using Linked Lists</b> <ul style="list-style-type: none"> <li>- Provide code examples for implementing multiple queues using linked lists.</li> <li>- Compare the array and linked list implementations in terms of efficiency and complexity.</li> </ul> </li> </ol> </li> <li><b>3. Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>- Implement a simple system with multiple queues and perform operations on them.</li> </ul> </li> </ol>



<b>Closure</b>	<p>Recap the importance of multiple queues and their implementation methods.</p> <ul style="list-style-type: none"><li>• Suggested Reading</li></ul> <p>"Operating Systems: Internals and Design Principles" by William Stallings</p>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: In what scenarios are multiple queues more effective than a single queue?</li><li>• Quick quiz on the implementation and applications of multiple queues.</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept and functionality of priority queues.</li> <li>b. Implement priority queues using arrays and heaps.</li> <li>c. Analyze the time complexity of various operations in a priority queue.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples for priority queue operations</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Discuss the concept of a queue and its FIFO behavior.</li> <li>- Ask: What if some elements are more important than others?</li> <li>- Introduce the concept of priority queues and their importance in scheduling and other applications.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <p><b>a. Priority Queue Overview</b></p> <ul style="list-style-type: none"> <li>- Explain the structure and purpose of priority queues.</li> <li>- Discuss different types of priority queues (max-priority and min-priority).</li> <li>-</li> </ul> <p><b>b. Implementation Using Arrays</b></p> <ul style="list-style-type: none"> <li>- Provide code examples for implementing priority queues using arrays.</li> <li>- Discuss the time complexity of insertion, deletion, and retrieval operations.</li> <li>-</li> </ul> <p><b>c. Implementation Using Heaps</b></p> <ul style="list-style-type: none"> <li>- Explain the use of heaps for efficient priority queue operations.</li> <li>- Provide code examples for heap-based priority queues.</li> </ul> <p><b>3. Exercise (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Implement a priority queue using both array and heap methods.</li> </ul>
<b>Closure</b>	Summarize the lesson by discussing the significance of priority queues in



	<p>various applications.</p> <ul style="list-style-type: none"><li>• Suggested Reading</li></ul> <p>"Data Structures and Algorithm Analysis in C" by Mark Allen Weiss</p>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What is the advantage of using a heap for implementing a priority queue?</li><li>• Quick quiz on priority queue operations and their complexities.</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the concept and implementation of circular queues.</li> <li>Implement circular queues using arrays.</li> <li>Analyze the advantages of circular queues over linear queues.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Code examples for priority queue operations</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>Recap the structure and limitations of linear queues.</li> <li>Ask: What happens when a linear queue reaches its end?</li> <li>Introduce the concept of circular queues as a solution to the limitations of linear queues.</li> </ul> </li> <li><b>Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>Circular Queue Overview</b> <ul style="list-style-type: none"> <li>Explain the structure and working of circular queues.</li> <li>Discuss the advantages of circular queues over linear queues.</li> </ul> </li> <li><b>Implementation Using Arrays</b> <ul style="list-style-type: none"> <li>Provide code examples for implementing circular queues using arrays.</li> <li>Explain the wrap-around behavior and handling of front and rear pointers.</li> <li>Discuss potential issues like queue overflow and underflow.</li> </ul> </li> <li><b>Comparison with Linear Queues</b> <ul style="list-style-type: none"> <li>Analyze the scenarios where circular queues are more efficient than linear queues.</li> </ul> </li> </ol> </li> <li><b>Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>Implement a circular queue and perform basic operations like enqueue and dequeue.</li> </ul> </li> </ol>



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



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the Disadvantages of Arrays.</li> <li>Articulate the concept and uses of Linked List.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>Introduction (5 minutes)</b> Ask questions           <ul style="list-style-type: none"> <li>What do you understand by Structure?</li> <li>What is the need of structure?</li> <li>What is structure-pointer concept?</li> <li>Introduce the drawbacks of arrays.</li> </ul> </li> <li><b>Development (30 minutes)</b> <ol style="list-style-type: none"> <li>Disadvantages of Arrays               <ul style="list-style-type: none"> <li>wastage of memory.</li> <li>Slow Insertion/Deletion Time</li> </ul> </li> <li>Linked List Basics               <ul style="list-style-type: none"> <li>Definition</li> <li>Concept of nodes</li> <li>Representation</li> </ul> </li> <li>Uses of Linked list               <ul style="list-style-type: none"> <li>optimized utilization of space</li> <li>No advance declaration of memory size</li> <li>No Empty node</li> </ul> </li> <li>Why Linked List over array               <ul style="list-style-type: none"> <li>allocates the memory dynamically</li> <li>List grows as per the program's demand</li> </ul> </li> </ol> </li> <li><b>Exercise(5 minutes)</b></li> </ol>



	-Ask Students to why link list is preferable over arrays
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learning</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Identify different types of Linked List.</li> <li>b. Articulate how different types of linked list contribute to the data structures.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask questions What are the disadvantages of array?  What is Linked list?  What is node and how it is partitioned?</li> <li>- Introduce the different types of linked list.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. Classification of linked list <ul style="list-style-type: none"> <li>-Singly linked lists</li> <li>-Doubly linked lists</li> <li>-Circular linked lists</li> <li>-Circular doubly linked lists</li> </ul> </li> <li>b. Concept of different types of Linked List <ul style="list-style-type: none"> <li>-Singly linked lists (unidirectionallinked list)</li> <li>- Doubly linked lists (bi-directional linked list)</li> <li>- Circular Linked Lists (last node pointing to the head node)</li> <li>-Circular doubly linked lists (mixture of a doubly linked list and a circular linked list)</li> </ul> </li> </ul> <p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask students regarding the difference in different types of linked list.</li> </ul>



	Collect Responses with the help of Nearpod.
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> Spend 5 minutes to wrap up and consolidate the learnings.</li></ol>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Identify different types of Linked List.</li> <li>b. Articulate how different types of linked list contribute to the data structures.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask questions What are the disadvantages of array?</li> <li>What is Linked list?</li> <li>What is node and how it is partitioned?</li> <li>- Introduce the different types of linked list.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. Classification of linked list <ul style="list-style-type: none"> <li>- Singly linked lists</li> <li>- Doubly linked lists</li> <li>- Circular linked lists</li> <li>- Circular doubly linked lists</li> </ul> </li> <li>b. Concept of different types of Linked List <ul style="list-style-type: none"> <li>- Singly linked lists (unidirectional linked list)</li> <li>- Doubly linked lists (bi-directional linked list)</li> <li>- Circular Linked Lists (last node pointing to the head node)</li> <li>- Circular doubly linked lists (mixture of a doubly linked list and a circular linked list)</li> </ul> </li> </ul> <p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask students regarding the difference in different types of linked list.</li> </ul> <p>Collect Responses with the help of Nearpod.</p>



<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

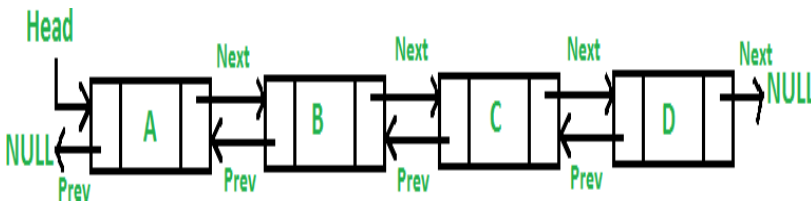
<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the logic behind the deletion of a node in a linked list.</li> <li>b. Gain knowledge about the types of deletions in a linked list.</li> <li>c. Articulate how nodes and links are updated while deletion process in a linked list.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- 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?</li> <li>- Introduce the concept of using deletion of nodes in a linked list.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. Introduction to deletion process. <ul style="list-style-type: none"> <li>- to check if list is empty or not.</li> <li>- to find out different modes of deletion in case of a linear linked list</li> </ul> </li> <li>b. Deletion in a linear linked list <ul style="list-style-type: none"> <li>- at the beginning of the linked list</li> <li>- At the end of the linked list</li> <li>- After a specific position</li> </ul> </li> <li>c. Code to delete a node linked list <ul style="list-style-type: none"> <li>- program to give a detailed explanation of deletion of a node in a linked list</li> </ul> </li> </ul> <p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask students about the deletion of node at 4<sup>th</sup> position in a linked list 5-&gt;10-&gt;15-&gt;20-&gt;25-&gt;30.</li> </ul>



<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

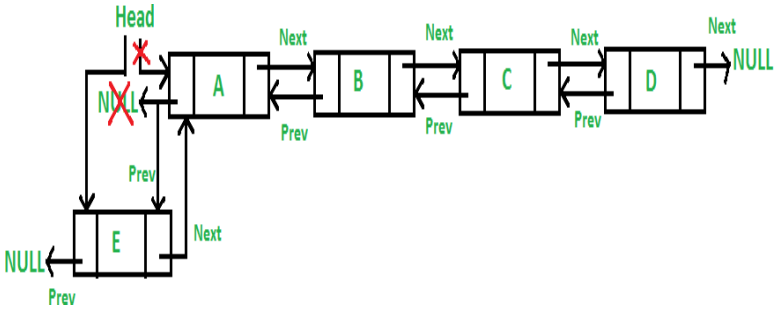
<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the structure a doubly linked list.</li> <li>Gain knowledge about the operations that can be performed in a doubly linked list.</li> <li>Articulate how nodes are traversed in a doubly linked list.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>Ask questions           <ul style="list-style-type: none"> <li>What is the structure of a single linked list?</li> <li>What is the difference between single and double linked list?</li> </ul> </li> <li>Discuss the advantages and disadvantages of double linked list over linear linked list</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ol style="list-style-type: none"> <li>Introduce the structure of double linked list           <ul style="list-style-type: none"> <li>Use of structure to create node</li> </ul> </li> <li>Declaration of Double Linked Lists           <ul style="list-style-type: none"> <li>first variable as data</li> <li>next as a pointer(keep the address of the next node)</li> <li>prev as a pointer(keep the address of the prev node)</li> </ul> </li> </ol>  <p><b>3. Exercise (5 minutes)</b></p> <ol style="list-style-type: none"> <li>Example of a double linked list           <ul style="list-style-type: none"> <li>program to give a detailed explanation of creation of double linked list.</li> </ul> </li> </ol>

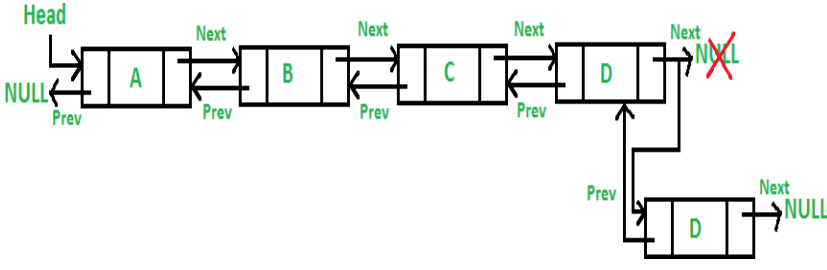
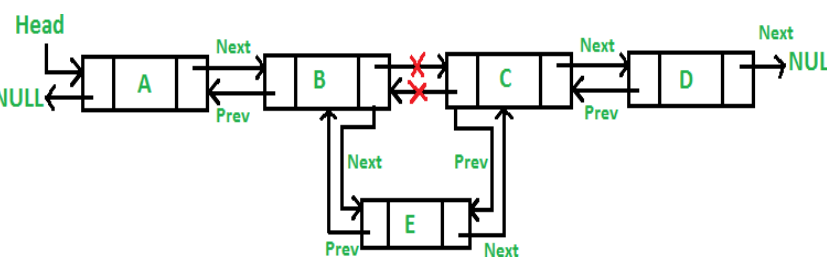


	<ul style="list-style-type: none"><li>- Ask students about the nodes and give them a program of traversal of a double linked list.</li></ul>
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li></ol> <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand the types of insertion operations that can be performed in a linked list.</li> <li>Gain knowledge about the updation of links of nodes in a linked list</li> <li>Articulate how nodes can be inserted at different position in a linked list.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Projector, Slides</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>Ask questions               <ul style="list-style-type: none"> <li>How can we traverse a doubly linked list?</li> <li>What is the difference between traversal and insertion?</li> </ul> </li> <li>Introduce the concept of insertion in a doubly linked list.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>Introduction to insertion               <ul style="list-style-type: none"> <li>to check if list is empty or not.</li> </ul> </li> <li>Insertion in a doubly linked list               <ul style="list-style-type: none"> <li>at the beginning of the linked list</li> </ul> </li> </ol> </li> </ol>  <ul style="list-style-type: none"> <li>At the end of the doubly linked list</li> </ul>

	 <p>- After a specific position in a doubly</p>  <p>c. Example to insert a node linked list - program to give a detailed explanation of insertion in a doubly linked list</p> <p>3. Exercise(5 minutes) - Ask students about the insertion and traversal of nodes and give them a program of insertion 30 at 4<sup>th</sup> position in a linked list 5,10,15,20,25,30.</p>
<p><b>Closure</b></p>	<ol style="list-style-type: none"> <li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li> <li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> Spend 5 minutes to wrap up and consolidate the learnings</li> </ol>
<p><b>Evaluation</b></p>	<ol style="list-style-type: none"> <li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li> <li>2. Nearpod Quiz on Data Structures</li> </ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the logic behind the deletion of a node in a doubly linked list.</li> <li>b. Gain knowledge about the types of deletions in a doubly linked list.</li> <li>c. Articulate how nodes and links are updated while deletion process in a doubly linked list.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask questions <ul style="list-style-type: none"> <li>How do we traverse a doubly linked list?</li> <li>What are the different modes of insertion?</li> <li>How can we count the number of nodes in a double linked list?</li> </ul> </li> <li>- Introduce the concept of using deletion of nodes in a double linked list.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. Introduction to deletion process. <ul style="list-style-type: none"> <li>- to check if list is empty or not.</li> <li>- to find out different modes of deletion in case of a double linked list</li> </ul> </li> <li>b. Deletion in a double linked list <ul style="list-style-type: none"> <li>- at the beginning of the linked list</li> <li>- At the end of the linked list</li> <li>- After a specific position</li> </ul> </li> <li>c. Code to delete a node linked list <ul style="list-style-type: none"> <li>- program to give a detailed explanation of deletion of a node in a double linked list</li> </ul> </li> </ul> <p><b>3. Exercise(5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask students about the deletion of node at 4<sup>th</sup> position in a linked</li> </ul>



	list 5->10->15->20->25->30.
<b>Closure</b>	<ol style="list-style-type: none"><li>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</li><li>2. Suggested Reading Online nptel course on data structures. <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a> Spend 5 minutes to wrap up and consolidate the learnings</li></ol>
<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What,Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Data Structures</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Code examples demonstrating memory leaks and garbage collection
<b>Teaching Development</b>	<b>1. Introduction (5 minutes)</b> - 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.  <b>2. Development (30 minutes)</b> <b>a. Garbage Collection Overview</b> - Define garbage collection and its role in memory management. - Discuss the problems caused by memory leaks and dangling pointers.  <b>b. Techniques of Garbage Collection</b>  • Explain different garbage collection techniques:  - <b>Reference Counting:</b> Describe how it works and its limitations (e.g., cyclic references). - <b>Mark and Sweep:</b> Explain the marking and sweeping phases. - <b>Generational Garbage Collection:</b> Discuss its efficiency in managing different object lifetimes.  <b>c. Implementation in C and Other Languages</b>  - Discuss manual memory management in C and the absence of built-in garbage collection. - Compare with languages like Java or Python, which have



	<p>automatic garbage collection.</p> <p><b>3. Exercise (5 minutes)</b></p> <ul style="list-style-type: none"><li>- Analyze a piece of code to identify potential memory leaks and discuss how garbage collection could address them.</li></ul>
<b>Closure</b>	<p>Summarize the importance of garbage collection in preventing memory leaks and improving program efficiency.</p> <ul style="list-style-type: none"><li>• Suggested Reading<ul style="list-style-type: none"><li>• "The Garbage Collection Handbook: The Art of Automatic Memory Management" by Richard Jones, Antony Hosking, and Eliot Moss</li><li>• Relevant chapters in "Operating Systems Concepts" by Silberschatz, Galvin, and Gagne</li></ul></li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: Why is garbage collection important? How do different garbage collection techniques work?</li><li>• Quick quiz on garbage collection concepts and techniques.</li></ul>



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Code examples demonstrating linked lists with and without header nodes
<b>Teaching Development</b>	<b>1. Introduction (5 minutes)</b> - 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.  <b>2. Development (30 minutes)</b> <b>a. Header Node Overview</b>  - 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>b. Implementation in Linked Lists</b> - 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.  <b>c. Applications and Use Cases</b> - Explore scenarios where header nodes are particularly beneficial, such as in circular linked lists or in algorithms where sentinel nodes are used.  <b>3. Exercise (5 minutes)</b>  -Implement a linked list with a header node and perform basic



	operations like insertion and deletion.
<b>Closure</b>	<p>Summarize the benefits of using header nodes in linked lists and how they simplify operations.</p> <p>Suggested Reading</p> <ul style="list-style-type: none"><li>• "Data Structures Using C" by Reema Thareja</li><li>• Relevant sections in "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein</li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What are the advantages of using a header node in linked lists? How does it affect the complexity of list operations?</li><li>• Quick quiz on header nodes and their implementatio</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of dynamic memory allocation and its importance in programming.</li> <li>b. Implement dynamic memory management using malloc(), calloc(), realloc(), and free() in C.</li> <li>c. Analyze common issues like memory leaks and fragmentation and how to handle them.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples demonstrating dynamic memory allocation and deallocation</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Discuss the limitations of static memory allocation.</li> <li>- Ask: What if the memory required by a program changes during its execution?</li> <li>- Introduce dynamic memory management as a solution to these limitations.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <p><b>a. Dynamic Memory Allocation Overview</b></p> <ul style="list-style-type: none"> <li>- Explain the need for dynamic memory allocation in programming.</li> <li>- Introduce the functions malloc(), calloc(), realloc(), and free() in C and their purposes.</li> </ul> <p><b>b. Implementation in C</b></p> <ul style="list-style-type: none"> <li>- Provide code examples demonstrating the use of malloc() and calloc() for dynamic memory allocation.</li> <li>- Discuss how realloc() is used to resize memory blocks, and the importance of free() in preventing memory leaks.</li> </ul> <p><b>c. Common Issues and Best Practices</b></p> <ul style="list-style-type: none"> <li>- Discuss common issues like memory leaks, dangling pointers, and memory fragmentation.</li> <li>- Provide best practices for managing dynamic memory, such as checking for NULL after allocation and always freeing allocated memory.</li> </ul> <p><b>3. Exercise (5 minutes)</b></p>



	<ul style="list-style-type: none"><li>- Write a program that dynamically allocates memory for an array, resizes it using <code>realloc()</code>, and then frees the memory.</li></ul>
<b>Closure</b>	<p>Recap the importance of dynamic memory management in writing efficient and flexible programs.</p> <p>Suggested Reading</p> <ul style="list-style-type: none"><li>• "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie</li><li>• Online resources on dynamic memory management in C, such as tutorials from GeeksforGeeks or the GNU C Library documentation.</li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What are the key differences between <code>malloc()</code> and <code>calloc()</code>? How do you prevent memory leaks in dynamic memory management?</li><li>• Quick quiz on dynamic memory allocation functions and their usage.</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of binary trees.</li> <li>b. Describe different types of binary trees and their properties.</li> <li>c. Implement binary trees using linked lists.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Discuss basic tree structures and their importance.</li> <li>- Ask: What is a binary tree? Why is it important?</li> <li>- Introduce the binary tree and its representation using linked lists.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <p><b>a. Concept of Binary Trees</b></p> <ul style="list-style-type: none"> <li>- Definition and properties</li> <li>- Types: Full, Complete, Perfect, and Degenerate Trees</li> </ul> <p><b>b. Representation Using Linked Lists</b></p> <ul style="list-style-type: none"> <li>- Node structure (data, left pointer, right pointer)</li> <li>- Creating a binary tree using linked list nodes</li> </ul> <p><b>c. Code Implementation</b></p> <ul style="list-style-type: none"> <li>- Provide and explain a sample code for binary tree creation</li> </ul> <p><b>3. Exercise (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Create a binary tree with nodes having values 1 to 5.</li> </ul>
<b>Closure</b>	<ul style="list-style-type: none"> <li>- Summarize the lesson and ensure understanding of binary tree representation.</li> <li>- Suggested Reading</li> <li>• "Data Structures and Algorithm Analysis in C" by Mark Allen Weiss</li> </ul>



<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What is a binary tree? How is it represented using linked lists?</li><li>• Quick quiz on binary tree concepts and representation.</li></ul> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Perform basic operations on binary trees.</li> <li>b. Implement insertion, deletion, and searching operations.</li> <li>c. Understand how these operations affect the structure of binary trees.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Discuss common operations on binary trees.</li> <li>- Ask: What operations do you think are crucial for binary trees?</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>a. Insertion Operation</b> <ul style="list-style-type: none"> <li>- Insert nodes into binary trees</li> <li>- Explain insertion logic with examples</li> </ul> </li> <li><b>b. Deletion Operation</b> <ul style="list-style-type: none"> <li>- Remove nodes and restructure the tree</li> <li>- Discuss different deletion scenarios</li> </ul> </li> <li><b>c. Searching Operation</b> <ul style="list-style-type: none"> <li>- Search for nodes in a binary tree</li> <li>- Explain search algorithms and complexity</li> </ul> </li> </ol> </li> <li><b>3. Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>• Implement insertion of nodes with values 6 to 10 in an existing binary tree.</li> </ul> </li> </ol>
<b>Closure</b>	<ul style="list-style-type: none"> <li>- Recap the operations discussed and their implementations.</li> <li>- Suggested Reading</li> <li>• "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and</li> </ul>



	Stein
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: How does insertion affect the binary tree? What happens during deletion?</li><li>• Quick quiz on binary tree operations.</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>a. Understand different binary tree traversal methods.</li> <li>b. Implement pre-order, in-order, and post-order traversals</li> <li>c. Analyze the efficiency of traversal algorithms.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples for traversal</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ol style="list-style-type: none"> <li>a. Explain the need for tree traversal.</li> <li>b. Ask: Why do we need different traversal methods?</li> </ol> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li>a. Explain the need for tree traversal.</li> <li>b. Ask: Why do we need different traversal methods?</li> <li><b>c. Traversal Methods</b> <ul style="list-style-type: none"> <li>- Pre-order traversal</li> <li>- In-order traversal</li> <li>- Post-order traversal</li> </ul> </li> <li><b>d. Code Implementation</b> <ul style="list-style-type: none"> <li>- Provide code for each traversal method</li> <li>- Explain code and output examples</li> </ul> </li> <li><b>c. Efficiency Analysis</b> <ul style="list-style-type: none"> <li>- Discuss time complexity of different traversals</li> </ul> </li> </ol> </li> <li><b>3. Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>- Implement and test all three traversal methods on a binary tree.</li> </ul> </li> </ol>
<b>Closure</b>	Summarize traversal methods and their use cases. <ul style="list-style-type: none"> <li>• Suggested Reading  "Algorithms" by Robert Sedgewick and Kevin Wayne</li> </ul>
<b>Evaluation</b>	<ul style="list-style-type: none"> <li>• Reflective Questions: What are the differences between traversal methods? How does traversal affect tree structure?</li> <li>• Quick quiz on traversal algorithms.</li> </ul>



Model Institute of Engineering  
& Technology (Autonomous)  
**Lesson Plan**

Kot, Bhalwal, Jammu



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1



Please Do Not Print Unless Necessary



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Identify real-world applications of binary trees</li> <li>b. Understand how binary trees are used in practical scenarios.</li> <li>c. Implement binary tree-based solutions to problems.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples for traversal</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Discuss the relevance of binary trees in various applications.</li> <li>- Ask: Can you think of any real-world applications of binary trees?</li> </ul> <p><b>2. Development (30 minutes)</b></p> <p><b>a. Applications Overview</b></p> <ul style="list-style-type: none"> <li>- Expression trees</li> <li>- Huffman coding trees</li> <li>- Binary search trees for databases</li> </ul> <p><b>b. Case Studies</b></p> <ul style="list-style-type: none"> <li>- Analyze examples of binary tree applications</li> <li>- Discuss advantages and limitations</li> </ul> <p><b>3. Exercise (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Solve a problem using a binary tree, such as constructing an expression tree.</li> </ul>
<b>Closure</b>	Recap the applications and discuss their impact. <ul style="list-style-type: none"> <li>• Suggested Reading</li> </ul> <p style="text-align: center;">"Computer Algorithms: Introduction to Design and Analysis" by Sara Baase and Alan Kleinberg</p>
<b>Evaluation</b>	<ul style="list-style-type: none"> <li>• Reflective Questions: What are the key applications of binary trees? How do they benefit specific use cases?</li> <li>• Quick quiz on applications of binary trees.</li> </ul>



Model Institute of Engineering  
& Technology (Autonomous)  
**Lesson Plan**

Kot, Bhalwal, Jammu



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1



Please Do Not Print Unless Necessary



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Code examples for traversal
<b>Teaching Development</b>	<b>1. Introduction (5 minutes)</b> - Explain the concept of threading in binary trees. - Ask: What challenges in tree traversal can be addressed by threading?  <b>2. Development (30 minutes)</b>  <b>a. Concept of Threaded Binary Trees</b> - Definition and types (single-threaded, double-threaded) - Benefits of threading  <b>b. Traversal Algorithms</b>  - Implement in-order traversal in threaded trees - Discuss other traversal methods and their efficiency  <b>3. Exercise (5 minutes)</b> - Implement threading in a binary tree and perform in-order traversal.
<b>Closure</b>	<ul style="list-style-type: none"><li>• Summarize threaded binary trees and their advantages.</li><li>• Suggested Reading  "Data Structures and Algorithms in C++" by Adam Drozdek</li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: How does threading improve tree traversal? What are the different types of threading?</li><li>• Quick quiz on threaded binary trees and traversal.</li></ul>



Model Institute of Engineering  
& Technology (Autonomous)  
**Lesson Plan**

Kot, Bhalwal, Jammu



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1



Please Do Not Print Unless Necessary



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of heterogeneous binary trees.</li> <li>b. Implement a heterogeneous binary tree structure.</li> <li>c. Analyze the use cases and limitations of heterogeneous binary trees.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples for traversal</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Define heterogeneous binary trees and their differences from homogeneous trees.</li> <li>- Ask: What scenarios require heterogeneous tree structures?</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>a. Concept and Implementation</b> <ul style="list-style-type: none"> <li>- Definition and structure</li> <li>- Example implementations in code</li> </ul> </li> <li><b>b. Use Cases and Limitations</b> <ul style="list-style-type: none"> <li>- Analyze where heterogeneous trees are useful</li> <li>- Discuss potential issues</li> </ul> </li> </ol> </li> <li><b>3. Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>- Implement a simple heterogeneous binary tree and explore its structure.</li> </ul> </li> </ol>
<b>Closure</b>	<ul style="list-style-type: none"> <li>• Recap the concept and applications of heterogeneous binary trees.</li> <li>• Suggested Reading <ul style="list-style-type: none"> <li>• "Data Structures and Algorithms" by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft</li> </ul> </li> </ul>
<b>Evaluation</b>	<ul style="list-style-type: none"> <li>• Reflective Questions: What makes a binary tree heterogeneous? How do you implement such a tree?</li> <li>• Quick quiz on heterogeneous binary trees.</li> </ul>



Model Institute of Engineering  
& Technology (Autonomous)  
**Lesson Plan**

Kot, Bhalwal, Jammu



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1



Please Do Not Print Unless Necessary



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand different ways to represent graphs.</li> <li>b. Implement adjacency matrix and adjacency list representations.</li> <li>c. Analyze the pros and cons of different representations.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Visual aids for graph representations</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Discuss what graphs are and their importance.</li> <li>- Ask: How do you think we can represent a graph in memory?</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>a. Adjacency Matrix Representation</b> <ul style="list-style-type: none"> <li>- Explain and provide examples</li> <li>- Discuss space and time complexity</li> </ul> </li> <li><b>b. Adjacency List Representation</b> <ul style="list-style-type: none"> <li>- Explain and provide examples</li> <li>- Compare with adjacency matrix</li> </ul> </li> <li><b>c. Other Representations</b> <ul style="list-style-type: none"> <li>- Discuss alternative methods like edge list</li> </ul> </li> </ol> </li> <li><b>• Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>• Implement a graph using both adjacency matrix and adjacency list.</li> </ul> </li> </ol>
<b>Closure</b>	<ul style="list-style-type: none"> <li>•Recap graph representations and their importance in various algorithms.</li> <li>• Suggested Reading <ul style="list-style-type: none"> <li>• "Graph Theory with Applications to Engineering and Computer</li> </ul> </li> </ul>



	Science" by Narsingh Deo
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What are the advantages of using an adjacency list over an adjacency matrix?</li><li>• Quick quiz on graph representation methods.</li></ul>



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Projector, Slides b. Code examples for DFS and BFS
<b>Teaching Development</b>	<b>1. Introduction (5 minutes)</b> - Explain the need for graph traversal. - Ask: Why do we need different traversal methods in graphs?  <b>2. Development (30 minutes)</b> <b>a. Depth First Search (DFS)</b> - Explain the algorithm and its working - Provide a code example and discuss complexity  <b>b. Breadth First Search (BFS)</b> - Explain the algorithm and its working - Provide a code example and discuss complexity  <b>c. Use Cases and Comparisons</b> - Analyze when to use DFS vs BFS  <b>3. Exercise (5 minutes)</b> - Implement both DFS and BFS on a sample graph.
<b>Closure</b>	<ul style="list-style-type: none"><li>• Summarize traversal methods and their applications in graph algorithms.</li><li>• Suggested Reading<ul style="list-style-type: none"><li>• "Introduction to Graph Theory" by Robin J. Wilson</li></ul></li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: How does DFS differ from BFS? When would you use one over the other?</li><li>• Quick quiz on DFS and BFS.</li></ul>



Model Institute of Engineering  
& Technology (Autonomous)  
**Lesson Plan**

Kot, Bhalwal, Jammu



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1



Please Do Not Print Unless Necessary



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of optimum search trees.</li> <li>b. Implement and analyze the construction of optimal binary search trees.</li> <li>c. Explore the use cases where optimum search trees provide efficiency.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples for constructing optimum search trees</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Discuss the importance of search efficiency in data structures.</li> <li>- Ask: What factors can affect the efficiency of search operations?</li> <li>- Introduce the concept of optimum search trees and their purpose.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>a. Optimum Search Trees Overview</b> <ul style="list-style-type: none"> <li>- Definition and significance</li> <li>- Difference between general and optimum search trees</li> </ul> </li> <li><b>b. Construction of Optimal Binary Search Trees</b> <ul style="list-style-type: none"> <li>- Dynamic programming approach</li> <li>- Explain cost functions and tree structure</li> <li>- Provide a step-by-step example of constructing an optimal binary search tree</li> </ul> </li> <li><b>c. Use Cases and Efficiency</b> <ul style="list-style-type: none"> <li>- Explore scenarios where optimum search trees are used</li> <li>- Discuss the time complexity and efficiency of search operations</li> </ul> </li> </ol> </li> <li><b>3. Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>- Construct an optimal binary search tree for a set of given probabilities and keys.</li> </ul> </li> </ol>



<b>Closure</b>	<ul style="list-style-type: none"><li>• Summarize the lesson by discussing the benefits of optimum search trees in improving search efficiency.</li><li>• Suggested Reading<ul style="list-style-type: none"><li>• "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein (Chapter on Dynamic Programming and Optimum Search Trees)</li></ul></li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What is the main advantage of using an optimum search tree? How is it constructed?</li><li>• Quick quiz on the concepts and construction of optimum search trees.</li></ul>



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand the concept of AVL trees and their properties.</li> <li>b. Implement AVL tree insertion and rotation operations.</li> <li>c. Analyze the balance factor and the need for balancing in binary search trees.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Projector, Slides</li> <li>b. Code examples for AVL tree operations</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>1. Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Introduce the concept of self-balancing trees.</li> <li>- Ask: What challenges do we face when a binary search tree becomes unbalanced?</li> <li>- Explain the need for AVL trees to maintain balance in binary search trees.</li> </ul> </li> <li><b>2. Development (30 minutes)</b> <ol style="list-style-type: none"> <li><b>a. AVL Trees Overview</b> <ul style="list-style-type: none"> <li>- Definition and properties</li> <li>- Balance factor and height of AVL trees</li> </ul> </li> <li><b>b. Insertion and Rotations</b> <ul style="list-style-type: none"> <li>- Explain the process of insertion in an AVL tree</li> <li>- Discuss different types of rotations (single and double) to maintain balance</li> <li>- Provide code examples for insertion and rotation operations</li> </ul> </li> <li><b>c. Balancing and Efficiency</b> <ul style="list-style-type: none"> <li>- Analyze the importance of balancing in AVL trees</li> <li>- Discuss time complexity for search, insertion, and deletion operations in AVL trees</li> </ul> </li> </ol> </li> <li><b>3. Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>- Insert a sequence of values into an AVL tree and demonstrate the necessary rotations.</li> </ul> </li> </ol>



<b>Closure</b>	<ul style="list-style-type: none"><li>• Summarize the lesson by discussing the significance of AVL trees in maintaining balanced binary search trees.</li><li>• Suggested Reading<ul style="list-style-type: none"><li>• "Data Structures and Algorithm Analysis in C" by Mark Allen Weiss (Chapter on AVL Trees)</li></ul></li></ul>
<b>Evaluation</b>	<ul style="list-style-type: none"><li>• Reflective Questions: What is the purpose of balancing in AVL trees? How do rotations help maintain balance?</li><li>• Quick quiz on AVL tree properties and operations.</li></ul>



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

<b>Objectives</b>	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.
<b>Teaching Aids (if any)</b>	a. Video of Facebook data center b. Use of Nearpod tool for online quiz
<b>Teaching Development</b>	<b>1. Introduction (5 minutes)</b> - 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. <b>2. Development (30 minutes)</b> 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.
<b>Closure</b>	Closure: 1. Summarize the Lesson Learning Outcomes and seek student affirmation. 2. Suggested Reading: - Online NPTEL course on data structures: <a href="#">NPTEL Course</a> Spend 5 minutes consolidating the learnings
<b>Evaluation</b>	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



Model Institute of Engineering  
& Technology (Autonomous)  
**Lesson Plan**

Kot, Bhalwal, Jammu



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1



Please Do Not Print Unless Necessary



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand and implement Selection Sort.</li> <li>b. Compare Selection Sort with Exchange Sort algorithms.</li> <li>c. Discuss the applications and limitations of Selection Sort.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Video of Facebook data center</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<p><b>1. Introduction (5 minutes)</b> Ask questions:</p> <ul style="list-style-type: none"> <li>- What are sorting algorithms?</li> <li>- How do Exchange Sort algorithms differ from other sorting methods?</li> <li>- Discuss the concept of Exchange Sort and its importance.</li> </ul> <p><b>2. Development (30 minutes)</b></p> <ul style="list-style-type: none"> <li>a. Bubble Sort <ul style="list-style-type: none"> <li>- Explain the algorithm and its process</li> <li>- Demonstrate Bubble Sort with a step-by-step example</li> <li>- Write a program to implement Bubble Sort</li> </ul> </li> <li>b. Quick Sort <ul style="list-style-type: none"> <li>- Explain the algorithm and its process</li> <li>- Discuss the concept of pivot and partitioning</li> <li>- Demonstrate Quick Sort with an example</li> <li>- Write a program to implement Quick Sort</li> </ul> </li> <li>c. Performance Comparison <ul style="list-style-type: none"> <li>- Compare Bubble Sort and Quick Sort in terms of time complexity and performance</li> </ul> </li> </ul> <p><b>3. Exercise (5 minutes)</b></p> <ul style="list-style-type: none"> <li>- Ask students to write programs for both Bubble Sort and Quick Sort and analyse their performance.</li> </ul>
<b>Closure</b>	<ol style="list-style-type: none"> <li>1. Summarize the Lesson Learning Outcomes and seek student affirmation.</li> <li>2. Suggested Reading: <ul style="list-style-type: none"> <li>- Online NPTEL course on data structures: <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li> <li>- Spend 5 minutes consolidating the learning.</li> </ul> </li> </ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>
<b>Evaluation</b>	<ol style="list-style-type: none"> <li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li> </ol>



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



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

<b>Objectives</b>	At the end of the lesson, students shall be able to: <ol style="list-style-type: none"> <li>Understand and implement Selection Sort.</li> <li>Compare Selection Sort with Exchange Sort algorithms.</li> <li>Discuss the applications and limitations of Selection Sort.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Video of Facebook data center</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>Ask questions:               <ul style="list-style-type: none"> <li>What is Selection Sort?</li> <li>How does it compare to Bubble Sort and Quick Sort?</li> </ul> </li> <li>Discuss the Selection Sort algorithm and its use cases.</li> </ul> </li> <li><b>Development (30 minutes)</b> <ol style="list-style-type: none"> <li>Selection Sort               <ul style="list-style-type: none"> <li>Explain the Selection Sort algorithm and its process</li> <li>Demonstrate Selection Sort with a step-by-step example</li> <li>Write a program to implement Selection Sort</li> </ul> </li> <li>Comparison with Exchange Sort               <ul style="list-style-type: none"> <li>Compare Selection Sort with Bubble Sort and Quick Sort in terms of time complexity and efficiency</li> </ul> </li> <li>Applications and Limitations               <ul style="list-style-type: none"> <li>Discuss scenarios where Selection Sort is beneficial and its limitations</li> </ul> </li> </ol> </li> <li><b>Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>Ask students to implement Selection Sort and compare its performance with Bubble Sort and Quick Sort</li> </ul> </li> </ol> <p>Use Nearpod to collect responses and discuss the answers.</p>
<b>Closure</b>	<ol style="list-style-type: none"> <li>Summarize the Lesson Learning Outcomes and seek student affirmation.</li> <li>Suggested Reading:       <ul style="list-style-type: none"> <li>Online NPTEL course on data structures: <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li> <li>Spend 5 minutes consolidating the learnings.</li> </ul> </li> </ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>



<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Selection sort</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
-------------------	---



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> <li>Understand and implement Insertion Sort.</li> <li>Compare Insertion Sort with other sorting methods.</li> <li>Analyze the performance and use cases of Insertion Sort.</li> </ol>
<b>Teaching Aids (if any)</b>	<ol style="list-style-type: none"> <li>Video of Facebook data center</li> <li>Use of Nearpod tool for online quiz</li> </ol>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li><b>Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>Ask questions:               <ul style="list-style-type: none"> <li>What is Insertion Sort?</li> <li>How does it differ from Bubble Sort and Quick Sort?</li> </ul> </li> <li>Discuss the concept of Insertion Sort and its typical use cases.</li> </ul> </li> <li><b>Development (30 minutes)</b> <ol style="list-style-type: none"> <li>Insertion Sort               <ul style="list-style-type: none"> <li>Explain the Insertion Sort algorithm and its process</li> <li>Demonstrate Insertion Sort with a step-by-step example</li> <li>Write a program to implement Insertion Sort</li> </ul> </li> <li>Comparison with Other Sorting Methods               <ul style="list-style-type: none"> <li>Compare Insertion Sort with Bubble Sort, Quick Sort, and Selection Sort in terms of performance and efficiency</li> </ul> </li> <li>Performance Analysis               <ul style="list-style-type: none"> <li>Discuss the time complexity and practical applications of Insertion Sort</li> </ul> </li> </ol> </li> <li>Exercise (5 minutes)           <ul style="list-style-type: none"> <li>Ask students to implement Insertion Sort and compare its performance with other sorting algorithms.</li> </ul> <p>Use Nearpod to collect responses and discuss the answers.</p> </li> </ol>
<b>Closure</b>	<ol style="list-style-type: none"> <li>Summarize the Lesson Learning Outcomes and seek student affirmation.</li> <li>Suggested Reading:           <ul style="list-style-type: none"> <li>Online NPTEL course on data structures: <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li> <li>Spend 5 minutes consolidating the learnings.</li> </ul> </li> </ol>



<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on insertion sort</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
-------------------	---



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

<b>Objectives</b>	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> <li>a. Understand and implement Merge Sort.</li> <li>b. Compare Merge Sort with other sorting algorithms such as Quick Sort and Bubble Sort</li> <li>c. Discuss the advantages and applications of Merge Sort.</li> </ul>
<b>Teaching Aids (if any)</b>	<ul style="list-style-type: none"> <li>a. Video of Facebook data center</li> <li>b. Use of Nearpod tool for online quiz</li> </ul>
<b>Teaching Development</b>	<ol style="list-style-type: none"> <li>1. <b>Introduction (5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask questions: <ul style="list-style-type: none"> <li>o What is Merge Sort?</li> <li>o How does Merge Sort work compared to other sorting algorithms?</li> </ul> </li> <li>- Discuss the concept and benefits of Merge Sort.</li> </ul> </li> <li>2. <b>Development (30 minutes)</b> <ul style="list-style-type: none"> <li>a. Merge Sort <ul style="list-style-type: none"> <li>- Explain the Merge Sort algorithm and its divide-and-conquer approach</li> <li>- Demonstrate Merge Sort with a step-by-step example</li> <li>- Write a program to implement Merge Sort</li> </ul> </li> <li>b. Comparison with Other Sorting Methods <ul style="list-style-type: none"> <li>- Compare Merge Sort with Quick Sort and Bubble Sort in terms of time complexity and performance</li> </ul> </li> <li>c. Advantages and Applications <ul style="list-style-type: none"> <li>- Discuss the advantages of Merge Sort, including its stability and efficiency in handling large data sets</li> </ul> </li> </ul> </li> <li>3. <b>Exercise (5 minutes)</b> <ul style="list-style-type: none"> <li>- Ask students to implement Merge Sort and analyze its performance compared to Quick Sort and Bubble Sort.</li> </ul> </li> </ol>
<b>Closure</b>	<ol style="list-style-type: none"> <li>1. Summarize the Lesson Learning Outcomes and seek student affirmation.</li> <li>2. Suggested Reading: <ul style="list-style-type: none"> <li>- Online NPTEL course on data structures: <a href="https://nptel.ac.in/courses/106105085">https://nptel.ac.in/courses/106105085</a></li> <li>- Spend 5 minutes consolidating the learnings.</li> </ul> </li> </ol> <p>Spend 5 minutes to wrap up and consolidate the learnings</p>



<b>Evaluation</b>	<ol style="list-style-type: none"><li>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</li><li>2. Nearpod Quiz on Merge sort</li></ol> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
-------------------	---