

Course Handout

Institute/School Name	Chitkara University Institute of Engineering & Technology		
Department Name	Department of Computer Science and Engineering		
Programme Name	B.E (Computer Science & Technology)		
Course Name	Data Structures using Object Oriented Programming	Session	July – Dec 2025
Course Code	24CSE0208	Semester/Batch	3 rd /2024
L-T-P(Per Week)	2-0-8	Course Credits	06
Pre-requisite	Basic concepts of computer fundamentals	NHEQF Level ¹	05 ¹
Course Coordinator	Dr. Heena Wadhwa	SDG Number ⁴	4, 8, 9 ⁴

1. Objectives of the Course

This course is an introduction to data structures using the concept of object oriented programming. It is to provide students with practical ability in the C++ language, which can be used for organizing and managing data for software development. The main objectives are:

- To apply core object-oriented programming principles to design and implement class hierarchies effectively
- To apply control structures to control the flow of program execution, facilitating efficient problem-solving and decision-making
- To develop proficiency in using data structures like arrays, functions and inheritance for effective data organization and manipulation
- To understand and implement fundamental data structures such as stacks, queues, linked lists, and trees for effective problem-solving
- To evaluate time-space complexity trade-offs in algorithm design and implement efficient searching and sorting techniques using data structures
- To design and optimize algorithms using data structures, tree traversal techniques, and graph operations for efficient data handling.

2. Course Learning Outcomes (CLOs)

Student should be able to:

	CLOs	Program Outcomes (PO)	NHEQF Level Descriptor ²	No. of Lectures
CLO01	Understand the principles of object oriented programming and implement basic programming for logic building by using data types, variables, operators and expressions	PO1, PO2, PO3, PO8	5.5	15
CLO02	Apply object-oriented programming constructs such as classes, objects, constructors, functions, pointers, recursion and inheritance to solve problems	PO1, PO2, PO4, PO5	5.5	30
CLO03	Determine the bugs in a program using exceptional handling and recognize basic need of templates	PO2, PO3, PO10, PO11	5.5	30
CLO04	Understand the basics of data structure, the complexity of algorithms, and the implementation of various operations on arrays and linked lists	PO1, PO2, PO3, PO9, PO11, PO12	5.5	20
CLO05	Implement searching and sorting algorithms for efficient data processing.	PO4, PO6, PO7, PO8, PO12	5.5	16
CLO06	Illustrate the concept of stack, linked list and queue data structures	PO1, PO2, PO5, PO6, PO11	5.5	22
CLO07	Analyze different tree traversal techniques and graph operations to model and address real-world scenarios	PO3, PO5, PO11, PO12	5.5	17
Total Contact Hours				150

¹ National Higher Education Qualification Framework Level, Refer to annexure

² NHEQF Level Descriptor, Refer to Annexure & [Learning outcomes descriptors for qualification for all levels on the NHEQF](#)

³Types of Assessments can be referred from Type of Assessments. Refer to Annexure.

⁴For SDG Mapping with Courses, PI refer [SDG Mapping policy for Courses](#)

CLO-PO Mapping

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Type of Assessment's ³
CLO01	H	M	M					L					Formative, Summative
CLO02	H	H		L	M								Formative, Summative
CLO03		M	M							M	L		Formative, Summative
CLO04	M	M	H						M		M	M	Formative, Summative
CLO05				M		M	M	L				H	Formative, Summative
CLO06	M	M		L		L					M		Formative, Summative
CLO07			M		H						M	L	Formative, Summative

H=High, M=Medium, L=Low

3. Recommended Books:

B01: Object Oriented Programming with C++' by E Balagurusamy, 6th Edition, Tata McGraw Hill.

B02: Object Oriented Programming in C++' by Robert Lafore, 4th Edition, Galgotia

B03: Data Structures Through C++, Yashavant Kanetkar, BPB Publications, 3rd Edition, 2019.

B04: Data Structure, Algorithms and Applications Using C++, S. Sahni, Sillicon Press, 2nd edition, 2005

B05 : Programming: Principles and Practice Using C++, by Bjarne Stroustrup, 2nd Edition, 2014

Reference Books:

RB01: Introduction to Algorithms by Thomas H. Cormen, The MIT Pressm 3rd Edition, 2001

RB02: Data Structures Using C and C++ by Langsam, Yedidyah, Tenenbaum, Aaron M., Pearson, 2nd edition, 2006

RB03: Objects, Abstraction, Data Structures and Design Using C++, Elliot B. Koffman, Paul A. T. Wolfgang, Wiley Publications, First Edition 2006

4. Other readings and relevant websites:

Serial No	Link of Journals, Magazines, websites and Research Papers
1.	http://www.cprogramming.com/tutorial/c++-tutorial.html
2.	http://www.cplusplus.com/doc/tutorial/
3.	http://www.tenouk.com/cncplusplusutorials.html
4.	https://algs4.cs.princeton.edu/home/
5.	https://nptel.ac.in/courses/106105151
6.	https://nptel.ac.in/courses/106102064
7.	https://library.chitkara.edu.in/subscribed-books.php

5. Recommended Tools and Platforms

- Visual Studio IDE
- Tespad

6. Course Plan: Theory+ Lab Plan

Theory Plan

Lect. No.	Topic(s)
01-02	Introduction to basic concepts of object-oriented programming, key features, uses, and importance. Comparison between the procedural programming paradigm and object-oriented programming paradigm
03-04	Setting up your Environment, Installing C++, Code Editors / IDEs
05-07	Basic Operation- Arithmetic Operators, Logical operators, Bitwise operators, Control Flow & Statements
08-11	Defining Structure and classes. Introduction to Name Space, Specifying a class, Creating class objects, Accessing class members
12-18	Inheritance: introduction, defining derived classes, forms of inheritance(single, multilevel, multiple), Access specifiers – public, private, and protected (Explain with inheritance)
19-23	Static members, Static objects, constant member function, constant objects, Friend functions, Friend class, Need for constructors and destructors
24-30	Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Raw pointer , Smart Pointer
31-36	Polymorphism: Understanding polymorphism in C++; types – compile-time (function and operator overloading) and run-time (using virtual functions);
37-42	Functions in c++ :-Function overloading, call by reference, call by value , call by pointer, return by reference, Lambda functions, Recursion in C++. Abstract classes

ST-1	
43-47	Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism
48-50	Type conversion: Basic type to class type, Class type to basic type, Class to class type, Type casting
51-55	Standard Template Library: Containers, iostream, Iterators and Algorithms, Template Specialization.
56-59	Data Structures and Algorithms: Importance in programming and real-world applications, Elementary Data Organization, Data Structure Types and Operations Types: Linear vs Non-linear, Static vs Dynamic
60-64	Arrays:- 1d, 2d, multi dimensional array, Accessing address of a variable, matrix operations
65-71	Searching: Linear and Binary Search with their Complexity Analysis
72-78	Sorting techniques: Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Bubble sort, Heap sort
79- 84	Introduction of Stack: Basic Operations, Implementation of stack
85-91	Implementation of Queue: Queue I: Basic Understanding of queue and its implementation, Queue II: Linear, circular, priority queue, dequeue
92-95	Algorithm: Complexity Analysis, Time vs Space trade-offs, Asymptotic Notations for Complexity(Ω , ω , θ , O , o) Analysis
ST-2 (ST1 syllabus also included)	
96-104	Linked List: Introduction & its memory representation, Linked Lists I: Fundamentals Singly Linked List and its operations (Insertion, deletion, traversal, searching), Linked Lists II: Types and Structures Doubly Linked List, Circular linked list, Linked Lists III: Advanced Operations and Applications (merging, reversing, detecting, finding the intersection point of two linked lists)
105-112	Trees: Binary trees, complete binary trees, Binary Search Trees, Representing binary trees, Tree Traversal: preorder, In order, Post order and their algorithms, Insertion, Deletion and Searching of elements in Binary Trees, Practice Problems
113-120	Graphs: Basic terminology, directed and undirected graphs, Representation of graphs, Operations on Graph, Graph traversals: DFS, BFS and their implementation.
ST-3 (Project Based Evaluation)	
End Term Examination (ETE-Complete Syllabus)	

7. Delivery/Instructional Resources Theory Plan:

Lect No.	Topics	CLO	Book No, CH No, Page No	TLM	ALM	Web References	Audio-Video
01	Introduction to basic concepts of object-oriented programming, key features, uses, and importance. Comparison between the procedural programming paradigm and object-oriented programming paradigm	1	B1 Chapter 1 B2 Chapter 1	Lecture Discussion Demonstration method using a simulation or a tool	Assignment	http://www.cplusplus.com/doc/tutorial/	https://onlinecourses.nptel.ac.in/noc24_cs125/preview
02	Setting up your Environment, Installing C++, Code Editors / IDEs	1, 2	B5 Chapter 2	Discussion Demonstration method using a simulation or a tool	Lab Challenge Assignment	https://code.visualstudio.com/docs/setup/windows	https://code.visualstudio.com/docs/cpp/config-mingw
03-04	Basic Operation- Arithmetic Operators, Logical operators, Bitwise operators, Control Flow & Statements	1, 2	B1 Chapter 2,3 B2 Chapter 2	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.cplusplus.com/doc/tutorial/	https://nptel.ac.in/courses/106105151
05	Defining Structure and classes. Introduction to Name Space, Specifying a class, Creating class objects, Accessing class members	2	B1 Chapter 4 B2 Chapter 3,4	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.cplusplus.com/tutorial/cplusplus-tutorial.html	https://nptel.ac.in/courses/106105151
06-07	Inheritance: introduction, defining derived classes, forms of inheritance(single, multilevel, multiple), Access specifiers – public, private, and protected (Explain with inheritance),	2	B1 Chapter 6 B2 Chapter 9,10	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.cplusplus.com/tutorial/cplusplus-tutorial.html	https://nptel.ac.in/courses/106105151
08	Static members, Static objects, constant member function, constant objects, Friend functions, Friend class, Need for constructors and destructors	2,3	B1 Chapter 5 B2 Chapter 6, 7	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.cplusplus.com/tutorial/cplusplus-tutorial.html	https://nptel.ac.in/courses/106105151
09	Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Raw pointer, Smart Pointer	2,3	B1 Chapter 8 B2 Chapter 5,14	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.cplusplus.com/tutorial/cplusplus-tutorial.html	https://nptel.ac.in/courses/106105151

10-11	Polymorphism: Understanding polymorphism in C++; types – compile-time (function and operator overloading) and run-time (using virtual functions);	2,3	B1 Chapter 7,11 B2 Chapter 11	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.tenouk.com/cncplustutorials.html	https://nptel.ac.in/courses/106105151
12	Functions in c++ :-Function overloading, call by reference, call by value , call by pointer, return by reference, Lambda functions, Recursion in C++. Abstract classes	2,3	B1 Chapter 3,9 B2 Chapter 5,6	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.tenouk.com/cncplustutorials.html	https://nptel.ac.in/courses/106105151
ST-1							
13	Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism	3	B1 Chapter 13 B2 Chapter 14	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.tenouk.com/cncplustutorials.html	https://nptel.ac.in/courses/106105151
14	Type conversion: Basic type to class type, Class type to basic type, Class to class type. Type casting	1	B1 Chapter 12 B2 Chapter 13	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.tenouk.com/cncplustutorials.html	https://nptel.ac.in/courses/106105151
15-16	Standard Template Library: Containers, iostream, Iterators and Algorithms, Template Specialization.	3	B1 Chapter 14 B2 Chapter 15 B4 Chapter 2	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	http://www.cplusplus.com/doc/tutorial/	
17	Data Structures and Algorithms: Importance in programming and real-world applications, Elementary Data Organization, Data Structure Types and Operations Types: Linear vs Non-linear, Static vs Dynamic	4	B3 Chapter 1 B4 Chapter 1	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://algs4.cs.princeton.edu/home/	https://nptel.ac.in/courses/106102064
18	Arrays:- 1d, 2d, multi dimensional array, Accessing address of a variable, matrix operations	4	B1 Chapter 2 B2 Chapter 2	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://algs4.cs.princeton.edu/home/	https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-2-data-structures-and-dynamic-arrays/
19	Searching: Linear and Binary Search with their Complexity Analysis	4,5	B3 Chapter 4 B4 Chapter 3	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://algs4.cs.princeton.edu/home/	https://nptel.ac.in/courses/106102064
20	Sorting techniques: Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Bubble sort, Heap sort	4,5	B3 Chapter 5 B4 Chapter 4	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://algs4.cs.princeton.edu/home/	https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-5-linear-sorting/
21	Introduction of Stack: Basic Operations, Implementation of stack	6	B3 Chapter 6 B4 Chapter 5	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://algs4.cs.princeton.edu/home/	https://nptel.ac.in/courses/106102064
22-23	Implementation of Queue: Queue I: Basic Understanding of queue and its implementation, Queue II: Linear, circular, priority queue, dequeue	6	B3 Chapter 7 B4 Chapter 6	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://algs4.cs.princeton.edu/home/	https://nptel.ac.in/courses/106102064
24	Algorithm: Complexity Analysis, Time vs Space trade-offs, Asymptotic Notations for	4,5	B3 Chapter 9 B4 Chapter 1	Lecture Discussion Demonstration	Quiz Lab Challenge	https://ocw.mit.edu/courses/1-204-computer-	https://nptel.ac.in/courses/106102064

	Complexity(Ω , ω , θ , O , o) Analysis			method using a simulation or a tool	Assignment	algorithms-in-systems-engineering-spring-2010/8ee75d49f1cb9a947f1d3f15a2aa9e00MIT1_204S10lec05.pdf	
ST-2 (ST1 syllabus also included)							
25-26	Linked List: Introduction & its memory representation, Linked Lists I: Fundamentals Singly Linked List and its operations (Insertion, deletion, traversal, searching), Linked Lists II: Types and Structures Doubly Linked List, Circular linked list, Linked Lists III: Advanced Operations and Applications (merging, reversing, detecting, finding the intersection point of two linked lists)	6	B3 Chapter 8 B4 Chapter 7	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://cse.mait.ac.in/pdf/LAB%20MANUAL/DS.pdf	https://nptel.ac.in/courses/106102064
27-28	Trees: Binary trees, complete binary trees, Binary Search Trees, Representing binary trees, Tree Traversal: preorder, In order, Post order and their algorithms, Insertion, Deletion and Searching of elements in Binary Trees, Practice Problems	7	B3 Chapter 10 B4 Chapter 8	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://cse.mait.ac.in/pdf/LAB%20MANUAL/DS.pdf	https://ocw.mit.edu/courses/6-851-advanced-data-structures-spring-2012/resources/session-15-static-trees/
29-30	Graphs: Basic terminology, directed and undirected graphs, Representation of graphs, Operations on Graph, Graph traversals: DFS, BFS and their implementation.	7	B3 Chapter 11 B4 Chapter	Lecture Discussion Demonstration method using a simulation or a tool	Quiz Lab Challenge Assignment	https://cse.mait.ac.in/pdf/LAB%20MANUAL/DS.pdf	https://nptel.ac.in/courses/106102064
ST-3 (Project Based Evaluation)							
End Term Examination(ETE)							

Lab Plan:

Lab No.	Experiment	CLO	TLM	ALM	Web References	Audio-Video
01-02	Factorial of a number Swap two numbers without using third one Sum of a set of numbers	1	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/program-for-factorial-of-a-number/ https://www.geeksforgeeks.org/swap-two-numbers-without-third-variable-in-cpp/ https://www.geeksforgeeks.org/cpp-program-to-find-sum-of-first-n-natural-numbers/
03-04	Functions to Display String Print Kth largest number	1, 2	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.programiz.com/cpp-programming/strings https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/
05-07	Write a C++ program to declare a class . Declare pointer to class. Initialize and display the contents of the class members	1, 2	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/c-classes-and-objects/
08-11	Implementing inheritance in C++ Modes of Inheritance	2	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/inheritance-in-c/
12-18	Write a program to show the working of static member functions	2,3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/static-member-function-in-cpp/

			tool			
19-23	Concept of a Pointer, Declaring and Initializing Pointer Variables and Pointer Expressions	2,3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/c/c-pointers/
24-30	Constructors, Destructors & Friend Function	3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/friend-class-function-cpp/
31-37	Pointers in C++, Accessing the Address of a Variable:	2,3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/cpp-pointers/
38-41	Compile-time Polymorphism and Runtime polymorphism in C++	2,3	Demonstration method using a simulation NA or a tool	Lab Challenge	NA	https://www.wscubetech.com/resources/cpp/polymorphism
42-47	Function Overloading, Call by Value, Call by Reference Recursive Function and Lambda Function	2,3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/function-overloading-c/ https://www.geeksforgeeks.org/lambda-expression-in-c/
ST1						
48-50	Write a C++ program to create a custom exception.	3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/how-to-throw-custom-exception-in-cpp/
51-53	Demonstration to convert a class to another class type in C++	1	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/how-to-convert-a-class-to-another-class-type-in-c/
54-56	Demonstrating the use of various STL containers and algorithms.	3	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/containers-cpp-stl/
57-59	Operations of Data Structures	4	Demonstration method using a simulation or a tool	Lab Challenge	NA	
60-64	Find out the largest and second largest element in array. Reverse the element of an array.	4	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/dsa/program-to-find-largest-element-in-an-array/
65-71	Perform Linear Search and Binary Search on an array. Search the element by passing the array to a function and then returning the position of the element from the function	4,5	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/dsa/linear-search/
72-78	Sorting techniques: Selection Sort, Insertion Sort, Quick Sort, Merge Sort. Implementation of selection sort Implementation of Insertion sort Implementation of Quick sort	4,5	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/sorting-algorithms/

	Implementation of Merge sort					
79- 84	Create a stack and perform Pop, Push, Traverse operations on the stack using Linear Linked list	6	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/implement-a-stack-using-singly-linked-list/
85-91	The enqueue operation can be used to add the element to the rear of the queue. The dequeue operation can be used to remove the element from the front of the queues	6	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/dsa/introduction-and-array-implementation-of-queue/
92-95	Algorithm Complexity and Complexity Computation Calculate Time and Space complexity of algorithms	4,5	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/dsa/time-complexity-and-space-complexity/
ST2						
96-104	Create a linked list with nodes having information about a student and perform Insert a new node at specified position. Delete of a node with the roll number of student specified. Reversal of that linked list	6	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/student-record-management-system-using-linked-list/
105-112	Create a Binary Tree Perform Tree traversals (Preorder, Postorder, Inorder) Insertion, Deletion and Searching of element in Binary Tree	7	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/
113-120	Implementation of Graph Implementation of graph traversal algorithm (DFS,BFS)	7	Demonstration method using a simulation or a tool	Lab Challenge	NA	https://www.geeksforgeeks.org/cpp/implementation-of-graph-in-cpp/ https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/

8. Remedial Classes³

After every Sessional Test, identify weak learners, and prepare the student lists. The following methods would be used to improve the performance of students.

- Remedial Classes on Saturdays
- Offer supplementary materials or activities to reinforce concepts outside regular class hours.
- Special discussions would be planned and scheduled accordingly.

³ Refer to Annexure

9. Self-Learning⁴

Assignments to promote self-learning, survey of contents from multiple sources.

S.No	Topics	CLO	ALM	References/MOOCs
1	Creating a new file and read the content of file	3	Think – Pair- Share	https://www.geeksforgeeks.org/c/basics-file-handling-c/
2	Operations on Circular linked List and its applications	6	Think – Pair- Share	https://archive.nptel.ac.in/courses/106/102/106102064/
3	Dynamic programming to find the nth Fibonacci Number	5,6	Think – Pair- Share	https://www.w3schools.com/dsa/dsa_ref_dynamic_programming.php

10. Delivery Details of Content Beyond Syllabus⁵

Content beyond syllabus covered (if any) should be delivered to all students that would be planned, and schedule notified accordingly.

S.No	Advanced Topics, Additional Reading, Research papers and any	CLO	POs	ALM	References/MOOCs
1	Dijkstra algorithm	7	1, 2, 3, 4, 5, 10	Brainstorming session	https://www.w3schools.com/dsa/dsa_algo_graphs_dijkstra.php https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/

11. Evaluation Scheme & Components:

Assessment Type ⁶	Evaluation Component ⁷	Type of Component ⁸	No. of Assessments ⁹	% Weightage of Component	Max. Marks	Mode of Assessment	CLO
Formative	Component 1	Testpad module progress and completion #	--	05%	05	Online	1 - 7
Formative	Component 2	Formative Assessments (FA)	05§	10%	10	Online	1 - 7
Formative	Component 3	Sessional Tests (ST-1, ST-2)	02*	20%	20	Online	1 - 7
Summative	Component 4	ST-3 (Project-Based Evaluation)	01**	15%	15	Offline	1 - 4
Summative	Component 5	End Term Examinations	01***	50%	50	Online	1 - 7
Total			100%				

Any progress or attempt made on TestPad after the deadline will not be considered.

§ There will be 05 formative assessments, and the scores of the BEST 03 will be considered.

* Students will have to appear in all the Sessional Tests.

*Makeup Examination will compensate for either ST-1 or ST-2 (Only for genuine cases, based on the Dean's approval).

**ST-3 will be a project-based evaluation (No makeup exam will be taken for ST-3)

***As per Academic Guidelines minimum 75% attendance is required to become eligible for appearing in the End Semester Examination.

⁴ Refer to Annexure

⁵ Refer to Annexure

⁶ Refer to [Annexure 2 of NCrf](#)

⁷ Refer to Annexure

⁸ Refer to Annexure

⁹ Refer to Annexure

12. Syllabus of the Course:

Subject:			
S.No.	Topic(s)	No. of Lectures	Weightage %
1.	Introduction to basic concepts of object-oriented programming, key features, uses, and importance. Comparison between the procedural programming paradigm and object-oriented programming paradigm Setting up your Environment, Installing C++, Code Editors / IDEs Basic Operation- Arithmetic Operators, Logical operators, Bitwise operators, Control Flow & Statements. Defining Structure and classes. Introduction to Name Space, Specifying a class, Creating class objects, Accessing class members	11	8%
	Inheritance: introduction, defining derived classes, forms of inheritance(single, multilevel, multiple), Access specifiers – public, private, and protected (Explain with inheritance) Static members, Static objects, constant member function, constant objects, Friend functions, Friend class, Need for constructors and destructors	19	13%
	Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Raw pointer, Smart Pointer Polymorphism: Understanding polymorphism in C++; types – compile-time (function and operator overloading) and run-time (using virtual functions);	17	12%
	Functions in c++ :-Function overloading, call by reference, call by value, call by pointer, return by reference, Lambda functions, Recursion in C++. Abstract classes	12	7%
2	Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism Type conversion: Basic type to class type, Class type to basic type, Class to class type. Type casting ; Standard Template Library: Containers, iostream, Iterators and Algorithms, Template Specialization.	12	5%
	Data Structures and Algorithms: Importance in programming and real-world applications, Elementary Data Organization, Data Structure Types and Operations Types: Linear vs Non-linear, Static vs Dynamic Arrays:- 1d, 2d, multi dimensional array, Accessing address of a variable, matrix operations Searching: Linear and Binary Search with their Complexity Analysis	16	15%
	Sorting techniques: Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Bubble sort, Heap sort Introduction of Stack: Basic Operations, Implementation of stack	15	10%
	Implementation of Queue: Queue I: Basic Understanding of queue and its implementation, Queue II: Linear, circular, priority queue, dequeue Algorithm: Complexity Analysis, Time vs Space trade-offs, Asymptotic Notations for Complexity(Ω , ω , θ , O , o) Analysis	17	10%
	Linked List: Introduction & its memory representation, Linked Lists I: Fundamentals Singly Linked List and its operations (Insertion, deletion, traversal, searching), Linked Lists II: Types and Structures Doubly Linked List, Circular linked list, Linked Lists III: Advanced Operations and Applications (merging, reversing, detecting, finding the intersection point of two linked lists)	11	7%
3	Trees: Binary trees, complete binary trees, Binary Search Trees, Representing binary trees, Tree Traversal: preorder, In order, Post order and their algorithms, Insertion, Deletion and Searching of elements in Binary Trees, Practice Problems	9	6%
	Graphs: Basic terminology, directed and undirected graphs, Representation of graphs, Operations on Graph, Graph traversals: DFS, BFS and their implementation.	11	7%
15	End Term Exam		

13. Academic Integrity Policy:

Education at Chitkara University builds on the principle that excellence requires freedom where Honesty and integrity are its prerequisites. Academic honesty in the advancement of knowledge requires that all students and Faculty respect the integrity of one another's work and recognize the importance of acknowledging and safeguarding intellectual property. Any breach of the same will be tantamount to severe academic penalties.

This Document is approved by:

Designation	Name	Signature
Course Coordinator	Dr. Heena Wadhwa	
Head-Academic Delivery	Dr. Mrinal Paliwal	
Dean	Dr. Rishu Chhabra	
Date(DD/MM/YYYY)	25/06/2025	

Annexure

1. Pre-requisite

Mention The Pre-requisite skill set or course/s if it is expected to be studies before this course, otherwise write “not applicable”.

2. NHEQF levels

The NHEQF levels represent a series of sequential stages expressed in terms of a range of learning outcomes against which typical qualifications are positioned/located. NHEQF level 4.5 represents learning outcomes appropriate to the first year (first two semesters) of the undergraduate programme of study, while Level 8 represents learning outcomes appropriate to the doctoral-level programme of study.

Table 1: Higher education qualifications at different levels on the NHEQF

NHEQF level	Examples of higher education qualifications located within each level
Level 4.5	Undergraduate Certificate. Programme duration: First year (first two semesters) of the undergraduate programme, followed by an exit 4-credit skills-enhancement course(s).
Level 5	Undergraduate Diploma. Programme duration: First two years (first four semesters) of the undergraduate programme, followed by an exit 4-credit skills-enhancement course(s) lasting two months.
Level 5.5	Bachelor's Degree. Programme duration: First three years (Six semesters) of the four-year undergraduate programme.
Level 6	Bachelor's Degree (Honours/ Honours with Research). Programme duration: Four years (eight semesters).
Level 6	Post-Graduate Diploma. Programme duration: One year (two semesters) for those who exit after successful completion of the first year (two semesters) of the 2-year master's programme.
Level 6.5	Master's degree. (e.g. M.A., M.Com., M.Sc., etc.) Programme duration: Two years (four semesters) after obtaining a 3- year Bachelor's degree (e.g. B.A., B.Sc., B.Com.etc.).
Level 6.5	Master's degree. (e.g. M.A., M.Com., M.Sc., etc.) Programme duration: One year (two semesters) after obtaining a 4 -year Bachelor's degree (Honours/ Honours with Research) (e.g. B.A., B.Sc., B.Com. etc.).
Level 7	Master's degree.(e.g. M.E./M.Tech. etc.) Programme duration: Two years (four semesters) after obtaining a 4-year Bachelor's degree. (e.g. B.E./B.Tech. etc.)
Level 8	Doctoral Degree

3. NHEQF level descriptors

Each NHEQF level is structured based on the defined learning outcomes which lead to the expected graduate attributes/profile. The level descriptors reflect the expected outcomes of learning that should be achieved and demonstrated by graduates of a specific programme of study leading to a qualification at a specific NHEQF level.

Click [Learning outcomes descriptors for qualification for all levels on the NHEQF](#)

4. Course Outcomes

The number of Course Outcomes is recommended to be 4-5 for courses that do not contain practical component and 6 for those courses with a practical component. Flexibility can be sought by the post-graduate courses in this regard.

5. Theory/lab Plan

The following are the guidelines to be followed while creating plans

- Each session may be planned for a duration of 45/50mins (irrespective of the double hour or single hour scheduled in timetable).
- Every session must incorporate at least one active learning method which may or may not be part of the assessments.
- Put BoS Approved Syllabus in the topics. Deviations (if any) from BoS approved syllabus must be brought to the notice of BoS chairman & Dean Academics. After approval, revised handout should be submitted.
- The Topics elaborated in the Theory/Lab plan must match those in the course execution plan.

6. Teaching Learning Methods

The following are some of the Teaching & Learning methods that can be incorporated in session wise teaching learning plan.

• Teacher-centered Learning Methods:

- Lecture
- Discussion
- Demonstration method using a simulation or a tool
- Reviewing
- Questioning

• Learner-centered teaching & Learning methods:

- Active learning**, in which students solve problems, answer questions, formulate questions of their own, discuss, explain, debate, or brainstorm during class;
- Cooperative learning**, in which students work in teams on problems and projects under conditions that assure both positive interdependence and individual accountability; and
- Inductive teaching and learning**, in which students are first presented with challenges (questions or problems) and learn the course material in the context of addressing the challenges.

- iv. **Inductive methods** include inquiry-based learning, case based instruction, problem-based learning, project-based learning, discovery learning, and just-in-time teaching. It is important to integrate authentic, reflective and collaborative learning experiences when designing for student-centered learning.

7. Active Learning Methods

The following are some of the Active Learning Methods that can be incorporated in session wise teaching learning plan.

- One Minute Paper
- Group Discussion
- Student-Created PPT, Charts, Matrices, Flowcharts, Models
- The Fish Bowl
- Debate
- Video Synthesis
- Quiz/Test Questions
- Brain Storming Sessions
- Case Study
- Shadowing
- Leading Question
- Puzzle, Enigma, Contradiction
- Statement-Opinion-Summary
- Think / Pair / Share
- Peer Review
- Just in Time Teaching
- Statement-Opinion-Summary
- Peer Survey
- Focused Listing
- Role-Playing
- Student Field Work with Reflection
- Infusing Humor into Class Sessions
- Inviting Effective Guest Speakers

8. Remedial Classes

After every Sessional Test, identify weak learners, provide supplement course handout. Student list and Impact Observed report should be submitted to Dean through proper channel.

9. Self Learning

Plan 10% of topics in self-learning mode with discussions, ALM's and Assessment happening in the class.

10. Content Beyond Syllabus

Plan Advanced Topics, Experiments, Additional Reading, Research papers in self-learning mode with ALM's and Assessment happening in the regular class or lab. Usually caters advanced learners. Identify Advanced learners. For Extra classes, schedule should be notified accordingly.

11. Assessment Type

1. Assessment broadly can be classified into the following types:

- a. **Diagnostic assessments:** Diagnostic assessments are intended to help teachers identify what students know and can do in different domains to support their students' learning. These help teachers determine strengths of students in various areas to better address their specific needs.
- b. **Formative assessments:** Formative assessment refers to a wide variety of methods that teachers use to conduct in-process evaluations of student comprehension, learning needs, and academic progress during a lesson, unit, or a course. Formative assessments help teachers identify concepts that students are struggling to understand, skills they are having difficulty acquiring, or learning standards they have not yet achieved so that adjustments can be made to lessons, instructional techniques, and academic support.
- c. **Summative assessments:** Summative assessment is an assessment administered at the end of an instructional unit in a course. These assessments are intended to evaluate student learning by comparing performance to a standard or benchmark.
- d. **Ipsative assessments:** Ipsative assessment involves comparisons between past and current work to identify a learner's growth over time, rather than progress toward an external set of criteria. Therefore, Ipsative assessment is an internal or self-referenced assessment.
- e. **Norm-referenced assessments:** Norm-referenced tests report whether test takers performed better or worse than a hypothetical average student, which is determined by comparing scores against the performance results of a statistically selected group of test takers, typically of the same age or grade level, who have already taken the exam.
- f. **Criterion-referenced assessments:** Criterion-Reference tests measure the performance of test takers against the criteria covered in the curriculum.
- g. **Peer-to-Peer randomised Assessments:** Peers will be able to provide assessment in this case
- h. **Industry Validation of Effectiveness:** In the Vocation Education, Industry validation of effectiveness of training is particularly important.
- i. **Self-assessments:** To evaluate how much the learner has grasped by self-learning.

2. Other Assessment Methods: Conducting an assessment takes time, thought, attention, planning, and often collaboration. Each assessment tool, whether a short survey or detailed rubric, will be useful only insofar as it both addresses the outcomes well and is feasible to use.

- a. **Rubrics:** For assessing qualitative student work such as essays, projects, reports, or presentations. Rubrics serve well to clearly denote the specific expectations for an assignment, for collecting data for assessment of student learning outcomes, and for student performance. Rubrics can be used for grading, for providing feedback to students, and for informing and encouraging students to think about their own learning.
- b. **Portfolios and E-Portfolio:** Portfolios can provide a window into the process of student learning across a semester-long project that can be assessed (usually by using a rubric).
- c. **Curriculum Mapping:** A good curriculum map can serve to focus assessment, and the improvements that follow, where it will be most useful, informative, or effective.

d. **Structured Interviews:** While time-consuming, structured interviews are useful when specific questions need to be asked. It also leaves room for unplanned topics or ideas to emerge.

e. **Student Experience Surveys:** Student experience in research universities (SERU), including administration of on-line census SERU Undergraduate and Graduate Surveys, can yield important information about student perceptions and experiences.

12. Evaluation Component & Types

As per LMs we need to figure it out whether it is component 1, 2 or 3. In Types of Evaluation Component, we need to specify what type of evaluation we are performing like Continuous Evaluation or Sessional Test or End Term Examination.

13. No. of Assessments and Weightage of Components

Department will give guideline for number of assessments, mandatory or optional and weightage.