COURSE FILE

For

Data Structures using C++ (ECSE215L)

Faculty Name	Dr. Samya Muhuri Ms. Anika Dr. Ankur Gupta Dr. Aditya Bhardwaj Dr. Mohit Agarwal Dr. Vaibhav Pandey
CourseType	Foundation
Semester and Year	3rd Semester and 2ndYear
L-T-P	3-0-4
Credits	5
Department	Computer Science Engineering
Course Level	UG

SCHOOL OF ENGINEERING AND APPLIED SCIENCES

Department of Computer Science Engineering

Bennett University
Greater Noida, Uttar Pradesh

Detailed Syllabus of the Course

ECSE102L: Data Structures using C++

CourseType:	Foundation

L	T	P	Credits
3	0	4	5

- **Pre-requisites:** NA
- **Course Learning Outcomes:**

CLO1: Write recursive algorithms to handle all recursive data structures..

CLO2: Choose appropriate searching and sorting techniques and apply graph algorithms for various practical problems..

CLO3: Formulate new/improved solutions for programming problems using learned data structure.

Module 1 (Contact hours: 12)
Introduction to OOPS Concepts, handling Arrays, Tower of Hanoi; Binary search, Time Complexity, Asymptotic Analysis, Big –Oh notation, Solving Recurrence relations. Lists and Implementations: - Linked lists, Recursive functions on lists, Deletion, insertion, reversing, joining.

Module 2 (Contact hours: 12)
Stacks, Queues using linked lists, handling Polynomials; Tree data structure and Implementations, binary and complete binary trees; Tree traversals algorithms, Binary search trees, insertion and deletion.

Module 3 (Contact hours: 9)
Sorting Algorithms: - Merge and Quick sort; AVL trees, B-Trees, Heap Trees - Priority Queues, heap sort.

Module 4 (Contact hours: 9)
Hashing: -Chained Hash Tables, Linear Probing, Double Hashing; Disjoint Set Class: -Smart Union Algorithms, Path compression; Graph: -BFS, DFS Topologicalsort, Minimum spanning trees, Huffman coding, Shortest path, Tree data structure.

STUDIO WORK / LABORATORY EXPERIMENTS:

The laboratory of Data structures is designed to provide a practical exposure to the students about the concepts and topics taught in the classroom sessions.Implementing the learnt concepts using C++ will help the students to have a better understanding of the subject.

TEXTBOOKS/LEARNING RESOURCES:

a) Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C++ (2 ed.), Universities Press, 2008. ISBN 978-0929306377.
b) Langsem, Augenstein and Tenenbaum, Data Structures using C and C++Langsem, Augenstein and Tenenbaum (2 ed.), Pearson, 2015. ISBN 978-8120311770.

REFERENCE BOOKS/LEARNING RESOURCES:

a) A. V. Aho, J. E. Hopcroft, and J. D. Ullman, Data Structures and Algorithms (1 ed.), Pearson, 2002. ISBN 978-8178081021.

b) Aaron M. Tenenbaum, Y. Langsam and Moshe J. Augenstein, Data Structures Using C (2 ed.), Pearson, 2019. ISBN 978-8131703281.

- **MOOC:**

1) **Udemy:** The Coding Interview Bootcamp: Algorithms + Data Structures
https://www.udemy.com/topic/data-structures/?utm_source=adwords&utm_medium=udemyads&utm_campaign=Branded-Topic_la.EN_cc.INDIA&utm_content=deal4584&utm_term=_._ag_83677222041_._ad_536963903990_._kw_%2Bdata%20%2Bstructures%20%2Budemy_._de_c_._dm__._pl__._ti_kwd-646773113456_._li_9061835_._pd__._&matchtype=b&gclid=CjwKCAjw9uKIBhA8EiwAYPUS3AEhWp6w9xLKRdGmXwQt5dE8C2X0tnTExuF2FvU4Sgh1_Q57N8Ii9hoCwHIQAvD_BwE

2) Coursera: Data Science Specialization
https://www.coursera.org/specializations/jhu-data-science?utm_source=gg&utm_medium=sem&utm_campaign=03-DataScience-JHU-IN&utm_content=03-DataScience-JHU-IN&campaignid=12034932697&adgroupid=121414562332&device=c&keyword=data%20science%20course%20coursera&matchtype=b&network=g&devicemodel=&adpostion=&creativeid=490608671484&hide_mobile_promo&gclid=CjwKCAjw9uKIBhA8EiwAYPUS3C3a_56vRC2qfbFOCc77VWEc9z0vFXuw8mHgVPw2CT3bKH2SVr681RoCpDcQAvD_BwE

3) edX: Introduction to Data Structures
https://www.edx.org/course/introduction-to-data-structures

4) NPTEL: Data Structure and Algorithms
https://nptel.ac.in/courses/106/102/106102064/

Evaluation Component

Components of Course Evaluation	Percentage
Quiz	10
Mid Term	15
End Term	35
Lab Examination	15
Continuous Lab Evaluation	25
Total	**100**

Lecture and Lab Plan

WEEK	Lecture No.	Lecture Plan	Lab Plan	
W1	1	Introduction to course and C++ First Program.	Basic C++ Programs covering basics along with Simple Loops.	Basic C++ Programs having Classes, Objects, Abstraction and Encapsulation.
	2	Introduction to Basic OOPs Concepts [Classes, Objects, Abstraction, Encapsulation, Inheritance, Polymorphism] [Detailed Topic-wise Study with respect to C++: Self Study].		
	3	Handling Arrays: Introduction, Insertion, Deletion, Traversal, Search: Linear and Binary.		
W2	4	Recursion: Factorial and Tower of Hanoi.	Arrays Insertion, Deletion, Traversal and Searching.	Recursive Programs: Factorial, Tower of Hanoi, Binary Search *etc*.
	5	Handling Arrays: 1D and 2D Memory and Address Calculation.		
	6	Time Complexity: Asymptotic Analysis, Big –Oh notation.		
W3	7	Recurrence Relations: Substitution Method, Recursion Tree.	Basic C++ Programs on Inheritance, Polymorphism.	Introduction to Pointers and their usage via various programs.
	8	Recurrence Relations: Master Method.		
	9	Linked Lists: Introduction Singly Linked List: Traversal, Insertion, Deletion.		
W4	10	Circular Linked List: Traversal, Insertion, Deletion.	Singly Linked List: Insertion, Traversal, Deletion.	Circular Linked List: Insertion, Traversal, Deletion.
	11	Doubly Linked List: Traversal, Insertion, Deletion.		
	12	Linked Lists: Reversal, Polynomial Handling [Brief Introduction] [Joining of Linked Lists and Detailed Polynomial Handling : Self Study.]		
W5	13	Stacks: Introduction, Insertion, Deletion, Traversal.	Linked List: Polynomial Handling; Reversal and Joining.	Stacks: Insertion, Traversal, Deletion.
	14	Stacks Applications: Infix to Postfix Conversion, Post-fix Evaluation.		
	15	Queues: Introduction, Simple Queue : Insertion, Deletion, Traversal.		
	16	Circular Queue: Insertion, Deletion, Traversal and Priority Queues.	Stacks: Infix to Post Fix,	Simple Queues and Circular

W6	17	Trees as Data Structures with different types, Binary Search Trees Introduction, Three Traversals.	Post-fix Evaluation.	Queues: Insertion, Deletion, Traversal.
	18	Binary Search Tree: Insertion.		
W7	19	Binary Search Trees: Searching, Find Minimum, Maximum, Successor.	Priority Queues.	Binary Search Trees: Insertion.
	20	Binary Search Tree: Deletion.		
	21	AVL Trees: Introduction, Balance Factor, Rotations.		
W8	22	AVL Trees: Insertion.	Binary Search Trees: Traversals, Searching, Minimum, Maximum, Successor.	Binary Search Trees Deletion.
	23	AVL Trees: Deletion.		
	24	Sorting: Bubble Sort, Seletion Sort.		
W9	25	Sorting: Insertion Sort, Merge Sort.	Sorting: Bubble Sort, Selection Sort, Insertion Sort.	Sorting: Merge Sort.
	26	Sorting: Quick Sort.		
	27	Sorting: Counting Sort.		
W10	28	Sorting: Heap Sort.	Sorting: Counting Sort.	Sorting: Heap Sort.
	29	Priority Queues with respect to Heaps.		
	30	Industry Talk		
W11	31	Graphs: Introduction, Representations.	Graphs: Implementation.	Graphs Traversal: BFS.
	32	Graphs Traversal: BFS.		
	33	Graphs Traversal: DFS, Topological Sort.		
W12	34	Minimum Spanning Trees: Prim’s Algorithm.	Graphs Traversal: DFS.	Minimum Spanning Trees: Prim’s Algorithm.
	35	Disjoint Sets: Smart Union Algorithm, Path Compression.		
	36	Minimum Spaning Tree: Kruskal’s Algorithm.		
W13	37	Shortest Path: Dijkstra’s Algorithm	Minimum Spanning Trees: Kruskal’s Algorithm.	Shortest Path: Dijkstra’s Algorithm.
	38	Shortest Path: Floyd Warshall		
	39	Hashing: Introduction		
W14	40	Hashing Collision Resolution Strategies - I	Hashing Implementation.	Hashing Collision Resolution Strategies.
	41	Hashing Collision Resolution Strategies - II		
	42	Huffman Coding.		

- **List of tools (teaching and lab)**

1) Codezinger (for online coding/practical)
2) Mentimeter

- **Tentative Date for hackathon/longathon.**

Three hackathons are planned on last Saturday morning of each month (September, October, November).

- **Suggest at least 3 innovations how this will enhance learning outcome of the course:**

1. Using memtimeter to increase class engagement.
2. 3 hackathons are planned for helping students in placement.
3. Arrange industry talk to demonstrate students real-time problems.

- Youtube playlist
  https://www.youtube.com/channel/UCAYpK-_E9y1DzyOTZewl9dQ/playlists