COURSE OUTLINE- CS250 Data Structures and Algorithms

Department:	Faculty of Computing	**Knowledge Group:**	Programming Core	
Programme:	Computer Science	**Class:**	BSCS 13-C	
Course code:	CS-250	**Academic Session/Semester:**		Fall 2024
Course name:	Data Structures and Algorithms	**Pre/co requisite (course name and code, if applicable):**		CS-212 Object Oriented Programming
Credit hours:	3+1			

Course Synopsis	This course focuses on equipping students with a solid understanding of data structures and algorithms, which are foundational to software development. It aims to develop problem-solving skills by teaching students how to apply appropriate data structures and algorithms to real-world problems. Key topics include arrays, linked lists, stacks, queues, trees, graphs, sorting, searching algorithms, and their complexities.			
Course Learning Outcomes (CLOs)	At the end of the course the students will be able to: 1. Learn the basics of various data structures and algorithms, including their properties, use cases, and efficiency. They will be able to describe how these structures work and identify their applications. 2. Apply knowledge to solve real-world problems using suitable data structures and algorithms. They will learn to analyze the problem requirements and select the best solution approach. 3. Gain hands-on experience in programming using the latest Integrated Development Environments (IDEs). They will ensure that their code meets industry standards, focusing on testing, proper documentation, and packaging practices. 4. Familiarize themselves with the importance of ethical behavior and adherence to professional standards in software engineering, including respecting intellectual property, privacy, and industry norms. 5. Develop collaborative skills, working effectively both individually and within teams. They will learn to communicate their ideas, contribute to group discussions, and coordinate with peers to achieve common goals.			
Course Schedule	According to time table			
Course lecturer	**Name**	**Office**	**Contact no.**	**E-mail**
	Dr. Mehwish Fatima	201, IAEC Building		mehwish.fatima@seecs.edu.pk

Mapping of the Course Learning Outcomes (CLO) to the Programme Learning Outcomes (PLO), Teaching & Learning (T&L) methods and Assessment methods:

No.	Course Learning Outcomes	PLO (SE)	SO (CS/DS/AI)	BT Level	Teaching & Learning Methods	Assessment Methods
CLO 1	**Describe** the fundamentals of data structures and algorithms	NA	2	C-2 (Understand)	Active Leaning, Blended Learning	Quiz, Assignment, MSE, ESE
CLO 2	**Solve** a given real time problem by **applying** appropriate data structure and algorithm.	NA	3	C-3 (Apply)	Active Leaning, Cooperatinve Learning, Blended Learning	Quiz, Assignment, Lab, MSE, ESE

CLO 3	**Practice** programs using the latest IDEs ensuring testing, documentation and packaging of programs as per standards practices applicable to the software industry.	NA	5	P-3 (Guided Response)	Active Leaning, Cooperative Learning, Blended Learning	Lab
CLO 4	**Demonstrate** commitment to professional ethics by following engineering norms applicable to the software industry.	NA	7	A-3 (Valuing)	Active Leaning, Cooperative Learning	Project, Lab
CLO 5	**Contribute** effectively both individually and as a team member	NA	6	A-2 (Responding)	Active Leaning, Cooperative Learning	Project, Lab

Details on Innovative T&L practices:

No.	Type	Implementation
1.	Active learning	Conducted through in-class or lab activities.
2.	Cooperative learning	Conducted through lab activities and course projects to develop and document software using current IDEs, adhering to industry standards and professional ethics. Evaluations will include demos and observations for practical skills, and written reports for theoretical understanding.
3.	Blended learning	Conducted through the NUST Learning Management System (LMS), all information and materials related to teaching and learning activities will be shared with the class via this platform. Additionally, some formative assessments will be administered through the LMS.

Weekly Schedule:

Week 1	Introduction to Data Structures and Algorithms Recap: C++ Programming Review
Week 2	Arrays and Strings
Week 3	Linked Lists
Week 4	Applications of Stacks, and Queues
Week 5	Trees (Binary Trees)
Week 6	AVL Trees
Week 7	Run time complexity, function growth
Week 8	Merge sort
Week 9	**Mid-Semester Break**
Week 10	Linear Time Sorting, Recursion
Week 11	Quick Sort, Radix Sort
Week 12	Priority Queues
Week 13	Graphs
Week 14	Topological Sort
Week 15	Minimum Spanning Tree
Week 16	Shortest Path Algorithm
Week 17	Hash Tables
Week 18	**End Semester Break**

Lab Experiments (if applicable):

Lab 1	Implementation of pointers and dynamic memory allocation
Lab 2	Practice the concept of version control and test cases
Lab 3	Working with Multi-dimensional arrays
Lab 4	Implementation of linked list with its operations
Lab 5	Use knowledge of linked lists to implement a small functionality.
Lab 6	Implementation of stacks and queues in different problems
Lab 7	Implementation of three Sorting Algorithms and compare them
Lab 8	Implementation of Merge sort
Week 9	**Mid-Semester Break**
Lab 9	Concepts of Recursion, take a program using abstract stack and solve a popular puzzle
Lab 10	Implementation of Merge sort and Radix Sort Quick Sort
Lab 11	Execution of a code to represent the social network
Lab 12	Applied Data Structures: ChatBots Lab Activity or Bloom Filter, etc
Lab 13	Implementation of binary heap (min heap)
Lab 14	Implementation of Hash tables
Lab 15	Lab Exam
Lab 16	Project
Week 18	**End Semester Break**

Assessment Methods:

Assessment		**Percentage**
Theory: 75%		
1	Quizzes (10-15%)	10%
2	Assignments (5-10%)	10%
3	Mid-Term Exam (25-35%)	30%
4	End-Semester Exam (40-50%)	40%
5	Project Documentation	10%
Labs:25%		
1	Labs Tasks	70%
2	Final Lab	10%
3	Semester Project	20%
Total:		

Learning resources:

Textbook:

1. Data Structures & Algorithms Using C++, Fourth or latest Edition, Nell Dale

Reference Book:

1. Adam Drozdek. Data Structures and Algorithms in C++, sixth Edition (2016)
2. T. H. Cormen, Charles E. Leiserson, R. L. Rivest, Clifford S. Introduction to Algorithms, Third Edition (2009)
3. Mark A. Weiss, Data Structures and Algorithm Analysis in C++, Fourth Edition (2013)

4. Data Structures & Algorithms Using C++, Fourth or latest Edition, John Bullinaria (2019)

Grading Policy:

Quiz Policy:
The quizzes will be unannounced / announced and normally last for ten minutes. The question framed is to test the concepts involved in the last few lectures. Number of quizzes that will be used for evaluation is at the instructor's discretion.

Project Policy:
Students will be required to develop a project during the course which should be completed towards the end of the semester. They will be graded based on project deliverables and presentation at the end. Students will work in a group/team for projects. A group of 3 students is recommended. At most 4 students are allowed.

Assignment Policy:
In order to develop a comprehensive understanding of the subject, assignments will be given. Late assignments will not be accepted / graded. All assignments will count towards the total (No 'best-of' policy). The students are advised to do the assignment themselves. Copying of assignments is highly discouraged and violations will be dealt with severely by referring any occurrences to the disciplinary committee. The questions in the assignment are meant to be challenging to give students confidence and extensive knowledge about the subject matter and enable them to prepare for the exams.

Lab Conduct:
The labs will be conducted for three hours every week. A lab handout will be given in advance for study and analysis. The lab handouts will also be placed on LMS. The students are to submit their results by giving a lab report at the end of the lab for evaluation. One lab report per group will be required. However, oral exams shall also be conducted during lab sessions.

Plagiarism:
SEECS maintains a zero-tolerance policy towards plagiarism. While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own. Plagiarism occurs when the words, ideas, assertions, theories, figures, images, programming codes of others are presented as your own work. You must cite and acknowledge all sources of information in your assignments. Failing to comply with the SEECS plagiarism policy will lead to strict penalties including zero marks in assignments and referral to the academic coordination office for disciplinary action.

Late Days Policy:
To motivate students to dedicate ample time and effort to their assignments and projects, this course incorporates a Late Days policy. Each student will be granted a total of 5 Late Days, offering them the flexibility to use these days as needed for submitting assignments or projects beyond the specified deadlines. It will be the responsibility of each student to self-monitor their Late Day usage throughout the semester. In cases where a student exceeds the allocated 5 Late Days, a 2% penalty will be applied to their assignments and projects for each additional late day used, until a cumulative penalty of 0% is reached. It is essential to understand that there is no advantage in refraining from utilizing the available Late Days. Furthermore, if a student surpasses the 24-hour threshold (1 second or more), it will be counted as a full Late Day.