	Code & No:	CS 210
Data Structures	Credits:	<u>3 (3,1,1)</u>
	Pre-requisite:	CS 120
	Co-requisite:	None
	Level:	5

Course Description:

The purpose of this course is to provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. The main objective of the course is to teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter. This course is also about comparing algorithms and studying their correctness and computational complexity. This course offers the students a mixture of theoretical knowledge and practical experience using C++.

Course Aims:

1) Understanding of different data structures that are suitable for problems to be solved efficiently
2) Understanding of problem solving paradigm
3) Understanding of the design and analysis of algorithms based of different data structures
4) Understanding of the algorithms complexity for both iterative as well as for recursive approaches
5) Understanding of sorting and searching techniques
6) Implementing of data structures and algorithms
7) Understanding of how common computational problems can be solved efficiently on a computer.

Student Outcomes (SOs):

☒(a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

☒(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

☒(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

☒(d) An ability to function effectively on teams to accomplish a common goal

☐(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

☐(f) An ability to communicate effectively with a range of audiences

☐(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

☐(h) Recognition of the need for and an ability to engage in continuing professional development

☒(i) An ability to use current techniques, skills, and tools necessary for computing practice.

☐(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CS]

☐(k) An ability to apply design and development principles in the construction of software systems of varying complexity. [CS]

☐(j) An ability to use and apply current technical concepts and practices in the core information technologies of human computer interaction, information management, programming, networking, and web systems and technologies. [IT]

☐(k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems. [IT]

☐(l) An ability to effectively integrate IT-based solutions into the user environment. [IT]

☐(m) An understanding of best practices and standards and their application. [IT]

☐(n) An ability to assist in the creation of an effective project plan. [IT]

Course Learning Outcomes (CLOs):

1. To recognize the efficiency trade-offs of using arrays, hash tables, linked lists, heaps, and trees.
2. To recognize when a general collection, stack, queue, priority queue, or graph structure is required to solve a problem.
3. To be able to implement the insert, delete, and search operations on all the structures presented.
4. To be able for each data structure presented, to state in big O notation the running times.
5. To be able to analyze and choose the appropriate data structure in the design and implementation of small projects.

SOs and CLOs Mapping:

CLO/SO	a	b	c	d	e	f	g	h	i	j	k	l	m	n
CLO1	√	√	√						√					
CLO2	√	√	√						√					
CLO3	√	√	√						√					
CLO4	√	√							√					
CLO5			√	√										

No.	Topics	Weeks	Teaching hours
1	Introduction to data structures and principles of software engineering	1	3
2	Data Design and Implementation (algorithm analysis, growth of functions, ADTs)	1	3
3	Sorted and Unsorted lists	1	3
4	Stacks, Queues	1	3
5	Linked Lists (sorted, unsorted, stacks, queues)	2	6
6	Binary Search Trees	2	6
7	Heaps, Graphs (array and linked structures, DFS, BFS)	2	6
8	Sorting (selection, insertion, bubble, merge, quick)	2	6
9	Searching (linear, binary)	1	3
10	Hashing	1	3
	Total	**14**	**42**

Textbook:

- C++ Plus Data Structures, Nell Dale, Jones & Bartlett Learning; 6th edition (September 9, 2016)

Essential references:

- Data Structures and Other Objects Using C++ , Third Edition, Michael Main and Walter Savitch, Prentice Hall, 4th edition, 2010
- Data Structures, A Pseudocode Approach with C++, Author: Richard F. Gilberg. And Behrouz A. Forouzan. Brooks/Cole (Thomson Learning), 2001.
- Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, Robert Sedgewick. Addison-Wesley, 1998.