



**COURSE OUTLINE : CS/IS 212**

**D Credit – Degree Applicable**

**COURSE ID 005226**

**Cyclical Review: August 2020**

**COURSE DISCIPLINE :** CS/IS  
**COURSE NUMBER :** 212  
**COURSE TITLE (FULL) :** Advanced Data Structures  
**COURSE TITLE (SHORT) :** Advanced Data Structures

### **CATALOG DESCRIPTION**

CS/IS 212 is designed to provide a thorough coverage of data structures with data abstraction applied to a broad spectrum of practical applications. Students who take this course master the principles of programming as a tool for problem solving. Students solve practical problems in a computer-equipped laboratory using an object oriented programming language, such as JAVA. Some specific topics covered include hash tables, trees, persistent structures, indexed files, and databases.

Total Lecture Units: 3.00

Total Laboratory Units: 0.00

**Total Course Units: 3.00**

Total Lecture Hours: 54.00

Total Laboratory Hours: 0.00

Total Laboratory Hours To Be Arranged: 0.00

**Total Contact Hours: 54.00**

**Total Out-of-Class Hours: 108.00**

Prerequisite: CS/IS 211 or equivalent.



**ENTRY STANDARDS**

	Subject	Number	Title	Description	Include
1	CS/IS	211	Data Structures	Create computer programs using data structures such as arrays, records, strings, linked lists, stacks, queues, and hash tables;	Yes
2	CS/IS	211	Data Structures	create simple recursive functions and procedures;	Yes
3	CS/IS	211	Data Structures	explain how abstraction mechanisms aid in creating reusable software components;	Yes
4	CS/IS	211	Data Structures	create simple programs in an object-oriented programming language;	Yes
5	CS/IS	211	Data Structures	compare and contrast object-oriented analysis and design with structured analysis and design.	No

**EXIT STANDARDS**

- 1 Create computer programs solving more complex OOP problems;
- 2 explain more complex abstract data types such as trees, graphs, hash tables, and heaps;
- 3 explain queues, dequeues, and priority queues;
- 4 explain and program binary trees, full binary trees, and complete binary trees
- 5 explain 2-3 trees and n-trees and their advantages

**STUDENT LEARNING OUTCOMES**

- 1 design and develop a complex object oriented programs;
- 2 analyze and explain complex abstract data types;
- 3 apply use of trees and understanding of terminology

**COURSE CONTENT WITH INSTRUCTIONAL HOURS**

	Description	Lecture	Lab	Total Hours
1	Review of Basic Algorithms • Recursive solutions • Array Searching • File searching	9	0	9
2	Linked Lists • List implementations that use arrays • List implementations that link data • Inheritance and lists • Sorted lists	5	0	5
3	Stacks • The Abstract Data Type (ADT) stack • Simple application of the ADT stack • Applications utilizing Postfix and Infix expressions • The relationship between stacks and recursion	5	0	5



**COURSE OUTLINE : CS/IS 212**

**D Credit – Degree Applicable**

**COURSE ID 005226**

**Cyclical Review: August 2020**

4	Queues • Queues • Deques • Priority queues	5	0	5
5	Class Relationships • Inheritance revisited • Dynamic binding and abstract classes • Applications • Advantages of an objects-oriented approach	5	0	5
6	Trees • Terminology • The ADT binary tree • The ADT binary search tree • General trees	5	0	5
7	Advanced Implementation of Tables • Balanced search trees • Hashing • Data with multiple organizations	5	0	5
8	Graphs • Terminology • Graphs as ADT • Graph traversals • Applications of graphs	5	0	5
9	External Methods • External storage • Sorting data in an external file • External tables	5	0	5
10	Advanced Topics in Data Structures	5	0	5
				<b>54</b>

**OUT OF CLASS ASSIGNMENTS**

- 1 homework exercises (e.g. develop ADTs, explain advantages and disadvantages of solutions, etc.);
- 2 programming problems (e.g. programming trees, binary trees, full binary trees, and complete binary trees).

**METHODS OF EVALUATION**

- 1 midterm examinations and quizzes;
- 2 final examination.



**COURSE OUTLINE : CS/IS 212**

**D Credit – Degree Applicable**

**COURSE ID 005226**

**Cyclical Review: August 2020**

**METHODS OF INSTRUCTION**

- Lecture
- Laboratory
- Studio
- Discussion
- Multimedia
- Tutorial
- Independent Study
- Collaboratory Learning
- Demonstration
- Field Activities (Trips)
- Guest Speakers
- Presentations

**TEXTBOOKS**

<b>Title</b>	<b>Type</b>	<b>Publisher</b>	<b>Edition</b>	<b>Medium</b>	<b>Author</b>	<b>ISBN</b>	<b>Date</b>
Data Abstraction and Problem Solving with C++: Walls and Mirrors.	Required	New York: Addison Welsey.	7		Carrano, Frank	978-0134463971	2016