

CS 3350 : DATA STRUCTURES

<i>Semester Hours:</i>	3.0	<i>Contact Hours:</i> 3
<i>Coordinator:</i>	Venu Dasigi	
<i>Text:</i>	Data Abstraction and Problem Solving With C++: Walls and Mirrors, 7/E	
<i>Author(s):</i>	FRANK CARRANO & TIMOTHY HENRY	
<i>Year:</i>	2016	

SPECIFIC COURSE INFORMATION

Catalog Description:

Abstract data types including stacks, queues, lists, trees and graphs. Introduction to analysis of algorithms. Recursive searching and sorting algorithms. Adaptation and use of generic data structures and types. Functional concepts. Prerequisite: MATH 2220 or MATH 3220 and Grade of C or better in CS 2020.

Course type: **REQUIRED**

SPECIFIC COURSE GOALS

- I can solve computational problems using recursion.
- I can implement and apply stacks, queues, trees, and other custom data structures.
- I can create generic functions and classes.
- I understand algorithmic complexity (e.g. Big “O” notation).
- I can understand the relationship between data structures and algorithms.
- I can understand the design tradeoffs (e.g., code complexity and performance) in data structures and algorithms.

COMPUTER SCIENCE STUDENT OUTCOMES ADDRESSED BY THIS COURSE

- CS 1 Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions
- CS 2 Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline

- CS 3 Communicate effectively in a variety of professional contexts
- CS 6 Apply computer science theory and software development fundamentals to produce computing-based solutions

SOFTWARE ENGINEERING STUDENT OUTCOMES ADDRESSED BY THIS COURSE

- SE 1 An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- SE 3 An ability to communicate effectively with a range of audiences

LIST OF TOPICS COVERED

- Review of CS2020 concepts (common/example topics below) (0.5 weeks = 4%)
 - Function Parameters
 - Function Templates
 - Arrays (1D and 2D)
 - Linked Lists
- Recursion (1 week = 7%)
 - Direct/linear
 - Tail recursion
- Stacks & Queues (priority queue) (2 weeks = 14%)
 - Class Templates
 - Applications
- Intro to Analysis of Algorithms (1 week = 7%)
 - Calculating running time using primitive operations
 - Rates of Growth: $O(n)$, Time Complexity, Space Complexity
- Trees (3.5 weeks = 24%)
 - Binary Trees, Binary Tree Manipulation
 - Binary Search Trees
 - Heaps and Priority Queues, Balanced Trees
- Introduction to graphs (2 weeks = 14%)
 - Representation (adjacency matrix/list)
 - Breadth-first and depth-first search

- Shortest path
- Hash Tables (1.5 weeks = 11%)
 - Hash Functions (load factor)
 - Open Hashing, Closed Hashing
 - Collision resolution (linear probing, chaining)
- Sorting algorithms (1.5 weeks = 11%)
 - Insertion Sort, Selection Sort, Merge Sort, Quicksort, Heapsort
- Advanced programming concepts (.5 week = 4%)
 - Smart pointers
 - Exception handling (throw/catch)
- Adaptation and use of STL generic data structures, types, and algorithms (1 week = 7%)
 - Smart pointers
 - Iterators
 - Vectors

NOTE: Memory management themes (memory leaks, deep copy, etc.) are emphasized throughout.

COMPUTER SECURITY TOPICS

Faculty who recently offered CS 3350 have discussed and identified a list of topics related to computer security in this course. Below is a list for instructors to incorporate. (*) indicates topics that are mandatory.

Security Topic	Description	Textbook Reference ¹	Estimated Class Hours
*Initialization before use	Review how C++ default initialization occurs – what is/isn't initialized prior to use, why is that important?	Interlude 2	1
*Principle of least privilege	Making class data members private and allowing only certain 'holes in the wall' to access what the user needs and nothing more. Controlling what sub-types can access via private vs. protected in inherited classes.	Chapter 1 Interlude 1 Interlude 2	2
*Hash functions	Discussion of what a hash function is, several	Chapter 18	<1

¹Data Abstraction and Problem Solving With C++ by Carrano & Henry, 7th Edition.