



CIS 22C.73Z: CRN: 38591, Winter 2024

Data Abstraction and Structures

Credit-Degree Applicable

4.5 quarter units (3 semester units)

Class Meets Online Asynchronously, but you can attend the live streamed recording. Classes are recorded at 3:30 pm (PST) every Wed.

Office Hours in Zoom: TBA.

Instructor: Hussein Al-Hussein, Ph.D. (MS, Ph.D. from Stanford University)

Email: alhussein@fhda.edu

Course Registration Number (CRN): 38591

Section: 73Z

Units: 4.5 quarter units (3-semester units)

Welcome to your class [zyBook](#)

Textbook: CIS 22C: Data Abstraction and Structures

zyBook ISBN: 979-8-203-27820-3

Instructions for students

Students will access zyBooks through links in an LMS (Blackboard, Canvas, etc.)

1. Click any zyBooks assignment link in your learning management system
(Do not go to the zyBooks website and create a new account)
2. Subscribe

A subscription is **\$49.95**. Students may begin subscribing on Dec 25, 2023 and the cutoff to subscribe is Mar 21, 2024. Subscriptions will last until Apr 12, 2024.

Description from Catalog:

Application of software engineering techniques to the design and development of large programs; data abstraction and structures and associated algorithms: stacks, queues, linked lists, trees, graphs, and hash tables; internal and external sorting; use of recursion; team project.

Student Learning Outcome Statements (SLO):

- Read, analyze, and explain advanced data structures programs.
- Design solutions for advanced problems using appropriate design methodology incorporating advanced data structures programming constructs.
- Create and analyze the efficiency of advanced-level data structures algorithms, code, document, debug, and test advanced data structures programs using multiple source and header files.

Advisory preparation:

- CIS 22B or CIS 35A.
- Advisory: Mathematics 212 or equivalent.

Work Required: 15 hours per week

Grading:

- ZyBooks & Labs: 50%
- Midterm: 25%
- Final: 25%

Grade average required:

- A+: 98-100
- A: 92-97
- A-: 90-91
- B+: 88-89
- B: 82-87
- B-: 80-81
- C+: 78-79
- C: 70-77
- D+: 68-69
- D: 62-67
- D-: 60-61
- F: 59 and less

Student Resources:

The college has gathered all Canvas Resources for Students into a library; here is the link:

<https://deanza.instructure.com/courses/3382>

Expanded Description: Content and Form the zyBooks

ZyBook Sections

Table of contents (ZyBooks Sections)

Ch 01. Introduction to Data Structures and Algorithms

1.1 Data structures

1.2 Introduction to algorithms

1.4 Abstract data types

1.6 Algorithm efficiency

[1.7 LAB: Introduction to data structures labs](#)

Ch 02. Searching and Algorithm Analysis

- 2.1 Searching and algorithms
- 2.2 Binary search
- 2.3 C++: Linear and binary search
- 2.5 Growth of functions and complexity
- 2.6 O notation
- 2.7 Algorithm analysis
- 2.8 Recursive definitions
- 2.9 Recursive algorithms
- 2.10 Analyzing the time complexity of recursive algorithms
- [2.11 LAB: Binary search template function](#)

Ch 03. Sorting Algorithms

- 3.1 Sorting: Introduction
- 3.2 Selection sort
- 3.3 C++: Selection sort
- 3.4 Insertion sort
- 3.5 C++: Insertion sort
- 3.8 Quicksort
- 3.9 C++: Quicksort
- 3.10 Merge sort
- 3.11 C++: Merge sort
- 3.15 C++: Sorting with different operators
- [3.16 LAB: Natural merge sort](#)

Ch 04. Lists

- 4.1 List abstract data type (ADT)
- 4.2 Singly-linked lists
- 4.3 Singly-linked lists: Insert
- 4.4 Singly-linked lists: Remove
- 4.5 Linked list search
- 4.6 C++: Singly-linked lists
- 4.7 Doubly-linked lists
- 4.8 Doubly-linked lists: Insert
- 4.9 Doubly-linked lists: Remove
- 4.10 C++: Doubly-linked lists
- 4.11 Linked list traversal
- 4.12 Sorting linked lists
- 4.13 C++: Sorting linked lists
- 4.16 Array-based lists
- 4.17 C++: Array-based list
- [4.18 LAB: Sorted number list implementation with linked lists](#)

Ch 05. Stacks and Queues

- 5.1 Stack abstract data type (ADT)
- 5.2 Stacks using linked lists
- 5.3 C++: Array-based stacks
- 5.4 Queue abstract data type (ADT)
- 5.5 Queues using linked lists
- 5.6 C++: Array-based queues
- 5.7 C++: Stacks and queues using linked lists
- 5.8 Deque abstract data type (ADT)
- [5.9 LAB: Grocery list editor with undo stack](#)

Ch 06. Hash Tables

- 6.1 Hash tables
- 6.2 Chaining
- 6.3 Linear probing
- 6.5 Double hashing
- 6.6 Hash table resizing
- 6.7 Common hash functions
- 6.8 Direct hashing
- 6.10 C++: Hash tables
- [6.11 LAB: Course gradebook with unordered_map](#)

Ch 07. Trees

- 7.3 Binary search trees
- 7.4 BST search algorithm
- 7.5 BST insert algorithm
- 7.6 BST remove algorithm
- 7.7 BST inorder traversal
- 7.8 BST height and insertion order
- 7.10 BST: Recursion
- 7.12 C++: Binary search tree
- [7.13 LAB: BST validity checker](#)

Ch 08. Balanced Trees

- 8.1 AVL: A balanced tree
- 8.2 AVL rotations
- 8.3 AVL insertions
- 8.4 AVL removals
- 8.5 C++: AVL Trees
- [8.11 LAB: AVL tree Nth largest operation](#)

Ch 10. Sets

- 10.1 Set abstract data type
- 10.2 Set operations

10.3 Static and dynamic set operations
10.4 C++: Set implementation
[10.5 LAB: Implementing StaticSet using unordered_set](#)

Ch 11. Graphs

11.3 Graph representations: Adjacency lists
11.4 Graph representations: Adjacency matrices
11.5 Graphs: Breadth-first search
11.6 Graphs: Depth-first search
11.7 Directed graphs
11.8 Weighted graphs
11.9 C++: Graphs
11.10 C++: Breadth-first search
11.11 C++: Depth-first search
11.18 Minimum spanning tree
11.19 C++: Minimum spanning tree
11.20 All pairs shortest path
11.21 C++: All pairs shortest path
[11.22 LAB: Graph representations](#)

Assignments

- A. Reading: Required reading from the online interactive text
- B. Doing the homework zyBooks assignments online.

Compilers & IDE:

- **Windows & Mac:**

Visual Studio 2022: Community (Free):

<https://visualstudio.microsoft.com/downloads>

VSCode (Windows and Linux and Mac) (Free):

<https://code.visualstudio.com/download>

- **Mac:** Xcode, Neovim
- **Online Compiler:**

https://www.onlinegdb.com/online_c++_compiler (recommended, free)

https://www.tutorialspoint.com/compile_cpp_online.php

<https://www.programiz.com/cpp-programming/online-compiler/>

Useful Tutorials:

<https://www.geeksforgeeks.org/cpp-tutorial>

<https://www.geeksforgeeks.org/cpp-tutorial/>

<https://www.tutorialspoint.com/cplusplus/index.htm>

<https://thispointer.com/c11-tutorial>

Useful Interview Problems:

<https://interview.leetcode.com/interview/login/>

<https://www.hackerrank.com/domains/cpp>

C++ Uses:

<https://www.simplilearn.com/tutorials/cpp-tutorial/top-uses-of-c-plus-plus-programming>

<https://www.codingninjas.com/blog/2021/07/29/c-vs-java-vs-python-which-one-to-choose/>