Computer Engineering 12
Abstract Data Types and Data Structures

Spring 2012
Mondays, Wednesdays, and Fridays
10:30 am – 11:35 am and 1:00 pm – 2:05 pm

Instructor

Instructor:	Darren Atkinson
E-mail:	datkinson@scu.edu
Office hours:	Tuesdays 1:15–2:15 pm and Wednesdays 9:30–10:30 am
Office:	EC 245
Website:	http://www.cse.scu.edu/~atkinson/teaching/sp12/012/

Teaching Assistants

Teaching assistant:	Gavin Hagiwara
E-mails:	ghagiwara@scu.edu
Lab hours:	Mondays and Tuesdays, 2:15–5:00 pm

Teaching assistant:	Yiwen Zhang
E-mails:	yiwenzhang1@gmail.com
Lab hours:	Wednesdays and Thursdays, 2:15–5:00 pm

Textbooks

Required:	Gilberg and Forouzan, *Data Structures – A Pseudocode Approach with C*, Brooks/Cole.
Recommended:	Kernighan and Ritchie, *The C Programming Language*, 2nd edition, Prentice Hall.

Grading

Lecture

Midterm exams:	50%	(4/25 and 5/16)
Final exam:	50%	(6/11 or 6/13)

Laboratory

Lab attendance:	10%	(each and every week)
Programming projects:	90%	(4/8, 4/20, 5/4, 5/18, 6/1, 6/8)

Overview

Abstract data types:	sets, lists (including stacks and queues), priority queues, and maps (dictionaries)
Data structures:	arrays, hash tables, linked lists, trees (including search trees), heaps, and graphs
Algorithms:	searching and sorting (selection-based, insertion-based, and exchange-based)

Pathways

This course is associated with the *The Digital Age* Pathway. If you declare this Pathway, you may use a representative piece of work from this course in the Pathway Portfolio you will complete during your senior year.

Course Objectives

Students will be able . . .

1. To understand abstract data types, data structures, and algorithms.
2. To develop programming skills, including the basics of software development such as the use of existing code and the compile-debug-execute cycle.
3. To use mathematics to reinforce importance of mathematical skills.

Learning Outcomes

Students will . . .

1. Discuss the relationships between an abstract data type and a data structure.
2. Separate an abstract data type into an interface and an implementation.
3. Compare and contrast simple container data types (lists, sets, maps, priority queues).
4. Compare and contrast classic data structures (arrays, hash tables, linked-lists, trees, graphs).
5. Implement the classic data structures in a low-level language such as C.
6. Know the average-case and worst-case running times for common operations (insertion, deletion, retrieval, minimum, maximum) on the classic data structures.
7. Compare and contrast classic searching and sorting algorithms.

Policies

Disability Accommodation Policy

To request academic accommodations for a disability, students must be registered with Disabilities Resources located in Benson, room 216. If you would like to register with Disabilities Resources, please visit their office in Benson 216 or call (408) 554-4109. You will need to register and provide professional documentation of a disability prior to receiving academic accommodations.

Academic Integrity Policy

The University is committed to academic excellence and integrity. Students are expected to do their own work and to cite any sources they use. A student who is guilty of a dishonest act in an examination, paper, or other work required for a course, or who assists others in such an act, may, at the discretion of the instructor, receive a grade of F for the course.

In addition, a student found guilty of a dishonest act may be subject to sanctions up to and including dismissal from the University as a result of the student judicial process as described in the *Community Handbook*.

A student who violates copyright laws, including those covering the copying of software programs, or who knowingly alters official academic records from this or any other institution is subject to similar disciplinary action.

Exam Policy

You are free to attend either class period for lectures, but you must take the exams in the class period for which you are registered. There will be no exceptions to this policy.