CSCI 244 – Fall 2013
Data Structures and Analysis of Algorithms

Ben Coleman
coleman@cs.moravian.edu
PPHAC 214

Office Hours: MWF 2:20 - 3:00
T 9:00 - 11:30
or by appointment

Course Description

An examination of issues dealing with static and dynamic aggregates of data. Topics covered include logical characteristics of various data organizations, storage structures implementing structured data, design and implementation of algorithms to manipulate such storage structures, and classical applications of data structures. Representative data structures include stacks, queues, ordered trees, binary trees, and graphs. Both contiguous and linked storage implementations are considered and performance issues discussed.

Course Goals

Upon completion of this course, a successful student will be able to:

- Describe the strengths and limitations of linear data structures, trees, graphs, and hash tables.
- Select appropriate data structures for a specified problem.
- Describe classic algorithms for sorting data and for searching problem spaces.
- Select an appropriate problem-solving strategy for a specified problem.

Required Texts

In addition to the following text, supplementary readings will be given periodically during the semester.

- *Introduction to the Design and Analysis of Algorithms* by Anany Levitin

You should expect to spend about an hour before each class session working through the readings. This means reading the text for detail, studying the syntax for new language features, and working to learn vocabulary – not just skimming through the material before class.

Graded Material

- **Homework** – The goal of homework problems is for you to practice using the current course content and to explore the topics in more detail. Problems will be assigned nearly every class session and will be due for the next problem session. See the course outline, below, for dates of problem sessions.

  During a problem session we will go over the problems assigned since the last problem. For each problem, students will work with a partner to determine the appropriate grade:

  (3) "I got it" – The solution is perfect or near perfect.

  (2) "I mostly got it" – The solution has some errors or omissions but was headed in the right direction.

  (1) "I was far off" – The solution has serious errors or omissions, but a serious attempt was made.

  (0) "I got nothing" – The solution shows little progress or the problem was not attempted.

  At the end of the semester, your homework grade will be computed as follows:

≥ 2.5	A
≥ 2	B
≥ 1.5	C
≥ 1	D
< 1	F

- **Tests** – Two tests will be given during the semester on Wednesday, October 2 and Wednesday, November 11. You may only re-schedule a test for college-approved absences or documented illness. In either case, you must contact me *before* the beginning of the test.

- **Programming Assignments** – Various programming assignments will be assigned during the semester. In some instances, you will simply implement a small stand-alone program. At other times, a sequence of assignments will build upon each other to produce a final program. All programming assignments will be graded based on correctness and the quality of testing.

- **Final** – The final will be cumulative and will be given in-class on Monday, December 9 at 1:30 p.m. Any change to the final exam schedule must be approved by both me and the dean of students.

Grade Determination

- (40%) Homework
- (25%) Tests
- (20%) Programming Assignments
- (15%) Final

All grades will be calculated on the standard scale using pluses and minuses.

Course Policies

- **Extensions** – Because homework problems are graded in-class, they cannot be accepted late or granted extensions. For programming assignments I am generous with extensions if you approach me *before* the day the assignment is due.
- **Absences** – Your attendance is expected at each class meeting, but I understand that students occasionally get sick, have obligations outside Moravian, and even over sleep. If you do miss class, please send me an email explaining your absence – preferably before the class session. Regardless of your reason for missing class, you are responsible for the contents of reading assignments, handouts, class activities, and class email.
- **Academic Honesty** – Except on tests, you are *encouraged* to discuss the material and work with other students in the course. Specifically, on homework and programming assignments you may discuss any portion of the assignment with your fellow students. This policy does not allow you to copy another student's work verbatim – you must produce your own code or write-up of the material. Work together to learn the concepts, but keep in mind that you are ultimately responsible for the material on the tests.
- **Disabilities** – Students who wish to request accommodations in this class for a disability should contact the assistant director of learning services for academic and disability support at 1307 Main Street, or by calling 610-861-1510. Accommodations cannot be provided until authorization is received from the Academic Support Center.

Course Outline

Date	Reading	Topic
M Aug 26	• 1.1[1] • 1.2 • 1.3	• Fundamentals of Algorithmic Problem Solving • Important Problem Types
W Aug 28	• Web Source	• Java
F Aug 30	• Web Source	• Java
M Sept 2		• Labor Day
W Sept 4	• Web Source	• Unit Testing
F Sept 6	• 1.4	• Fundamental Data Structures and Graph Implementations
M Sept 9		• Problem Session
W Sept 11	• 2.1[2] • 2.2	• Measuring Running Time • Asymptotic Notation
F Sept 13	• 2.3	• Analysis of Non-Recursive Algorithms

[1]Each time we begin a new chapter, you should read the text between the chapter heading and the first section.
[2]See previous note.

Date	Reading	Topic
M Sept 16	• Web Source	• Recursion
W Sept 18	• 2.4	• Analysis of Recursive Algorithms
F Sept 20		• Problem Session
M Sept 23	• 3.1[3] • 3.2 • 3.3	• Selection and Bubble Sorts • Sequential Search and Brute-Force String Matching • Closest-Pair and Convex-Hull
W Sept 25	• 3.4	• Exhaustive Search
F Sept 27	• 3.5	• Depth-First Search and Breadth-First Search
M Sept 30		• Problem Session
W Oct 2		• Test #1
F Oct 4	• 4.1[4] • 4.2	• Insertion Sort • Topological Sorting
M Oct 7	• 4.4 • 4.5	• Decrease-by-a-Constant-Factor Algorithms • Variable-Size-Decrease Algorithms
W Oct 9	• 5.1[5] • 5.2	• Merge Sort • Quick Sort
F Oct 11		• Sorting Wrap-Up and Project Overview
M Oct 14		• Fall Break
W Oct 16		• Problem Session
F Oct 18	• 5.3	• Binary Tree Traversals
M Oct 21	• 6.3[6]	• Balance Binary Search Trees
W Oct 23	• 6.4	• Heaps and Heap Sort
F Oct 25		• Problem Session
M Oct 28	• 7.1[7]	• Counting Sort • Radix Sort
W Oct 30	• 7.3	• Hashing
F Nov 1	• 8.1[8]	• Dynamic Programming
M Nov 4	• 8.3	• Optimal Binary Search Trees
W Nov 6	• 8.4	• Warshall's and Floyd's Algorithms

[3]See the previous note and the one before it.
[4]See the previous three notes.
[5]See the previous four notes.
[6]See the previous five notes.
[7]See the previous six notes.
[8]See the previous seven notes.

Date	Reading	Topic
F Nov 8		• Problem Session
M Nov 11		• Test #2
W Nov 13	• 9.1[9] • 9.2	• Prim's Algorithm • Kruskal's Algorithm
F Nov 15	• 9.3	• Dijkstra's Algorithm
M Nov 18	• 9.4	• Huffman Trees and Codes
W Nov 20		• Problem Session
F Nov 22	• 10.2[10]	• The Maximum-Flow Problem
M Nov 25 W Nov 27 F Nov 29		• Thanksgiving Break
M Dec 2	• 10.4	• The Stable Marriage Problem
W Dec 4		• Problem Session
F Dec 6		• Review

The details of this syllabus and schedule are subject to change based on our progress through the material.

[9] See the previous eight notes.

[10] See the previous nine notes. If you follow the directions of this note in a literal fashion, how many times will you be pointed to the first footnote? If you follow the directions all semester, how many times will you be pointed to the first footnote? What are the answers to these two questions if there are n chapters and there is a footnote on the first section of each chapter?