CSCI 244 – Fall 2015
Data Structures and Analysis of Algorithms

Ben Coleman
coleman@cs.moravian.edu
PPHAC 214

Office Hours: MW 2:30 - 3:30
R 10:00 - 12:00
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.

- *Data Structures and the Java Collections Framework, Third Edition* by William Collins

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 session. For each problem, grades will be assigned as:

  (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 average homework program score will be converted into your homework grade as follows:

≥ 2.5	A
≥ 2	B
≥ 1.5	C
≥ 1	D
< 1	F

- **Tests** – Two tests will be given during the semester on Monday, October 19 and Wednesday, December 2. 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 14 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

- (25%) Homework
- (25%) Tests
- (35%) 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 Academic and Disability Support in the Academic Support Center, Monocacy Hall, lower level, or by calling 610-861-1401. Accommodations cannot be provided until authorization is received from the Academic Support Center.

Course Outline

Date	**Reading**	**Topic**
M Aug 31		• Day 1 Activities
W Sept 2	• Chapter 0 • Chapter 1 • Chapter 2	• Java Review
F Sept 4	• Handout	• Unit Testing
M Sept 7	• `https://docs.oracle.com/javase/tutorial/essential/io/streams.html`	• Files and Exceptions
W Sept 9	• Chapter 4	• The Java Collections Framework • Netbeans and the Debugger
F Sept 11	• Chapter 3	• Algorithm Analysis
M Sept 14		• Algorithm Analysis
W Sept 16		• Problem Session
F Sept 18	• Chapter 5 pp. 155 – 191	• Recusion
M Sept 21		• Recursion
W Sept 23		• Problem Session

Date	Reading	Topic
F Sept 25	• Chapter 5 pp. 191 – 210	• Backtracking
M Sept 28	• Handout	• Dynamic Programming
W Sept 30		• Problem Session
F Oct 2	• Chapter 11 pp. 457 – 468	• Basic Sorts
M Oct 5	• Chapter 11 pp. 470 – 477	• Merge Sort
W Oct 7	• Chapter 11 pp. 477 – 489	• Quick Sort
F Oct 9	• Chapter 11 pp. 489 – 493	• Radix Sort
M Oct 12		• Fall Break
W Oct 14	• Chapter 11 pp. 468 – 470	• Theoretical Limitations of Sorting
F Oct 16		• Problem Session
M Oct 19		• Test #1
W Oct 21	• Chapter 6 • Chapter 7	• Arrays • Linked Lists
F Oct 23	• Chapter 8	• Stacks and Queues
M Oct 26		• Problem Session
W Oct 28	• Chapter 9 pp. 377 – 386	• Binary Trees
F Oct 30		• Recursion on Binary Trees
M Nov 2	• Chapter 9 pp. 386 – 393	• Binary Tree Traversals
W Nov 4	• Chapter 10 p. 401 – 430	• Binary Search Trees
F Nov 6		• Problem Session
M Nov 9	• Chapter 10 pp. 430 – 442	• AVL Trees
W Nov 11	• Chapter 12 pp. 501 – 525	• Red/Black Trees • Tree Maps
F Nov 13	• Chapter 13 pp. 551 – 590	• Heaps • Priority Queues • Heap Sort
M Nov 16		• Problem Session
W Nov 18	• Chapter 14 pp. 599 – 626	• Hashing
F Nov 20	• Chapter 14 pp. 636 – 635	• Collisions
M Nov 23		• Problem Session
W Nov 25 & F Nov 27		• Thanksgiving Break
M Nov 30	• Chapter 15 pp. 643 – 650	• Graphs
W Dec 2		• Test #2

Date	Reading	Topic
F Dec 4	• Chapter 15 pp. 650 – 659	• Breadth-First Search • Depth-First Search
M Dec 7	• Chapter 15 pp. 659 – 669	• Minimum Spanning Trees • Shortest Paths
W Dec 9		• Slip Day
F Dec 11		• Problem Session • Review

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