CSCI 241: Data Structures
Syllabus, Spring, 2017

Instructor: Geoffrey Matthews
Email: `geoffrey dot matthews at wwu dot edu`
Office hours: MTWF 10:00-10:50, CF 469

Lectures: MTWF 11:00-11:50, CF 227

Catalog copy: Design and implementation of hash tables, general trees, search trees, balanced trees and graphs. Comparison of sorting algorithms. Demonstration of the use of data structures in various applications. Evaluation of the best data structure for a particular task. Programming is required in implementation of concepts.

Goals: On completion of this course, students will demonstrate:

- Basic understanding of classic data structures including trees, hash tables and graphs.
- Basic understanding of various sorting algorithms.
- Thorough understanding of recursion in the definition of and operations on these data structures.
- The ability to select and design the proper data structures to problems requiring complex data structures.
- The ability to select the proper sorting algorithms for a problem.
- The ability to make judgments about the selected data structures for a problem.
- The ability to implement the introduced data structures and sorting algorithms.

Websites:

- For class materials: `https://github.com/geofmatthews/csci241`
- For turning in homework and grading: `https://wwu.instructure.com/`

Texts: The following texts will be used as resources, but no problems will be assigned from either book. Online reading will also be assigned as topics come up.

- *Building Java Programs: A Back to Basics Approach, 4th edition*, Reges and Stepp.

  You may already have this book from CSCI 145. Access to online content associated with the book is not required. This is an excellent introduction to Java, but does not cover all the material for CSCI 241.
- *Introduction to Algorithms, 3rd edition*, Cormen *et al.*

  You will need this book for CSCI 305 and CSCI 405, and it is an excellent resource on algorithms and advanced data structures.

Software: The labs already have Oracle's Java Development Kit (JDK) 7 as well as jGrasp (and other IDEs) installed for both Windows and Linux. If you would like to work on your own computer, you are welcome to download and install jGrasp `http://www.jgrasp.org/`.

Programming assignments: Programs must be turned in on canvas. Specifications must be followed exactly (do not rename classes or procedures because you think your names are better, do not change the signature of the methods, *etc.*). No late work will be accepted. It is the student's responsibility to make sure the *correct* file is submitted by midnight of the due date.

Quizzes: There will be a short quiz every Friday except the first and last weeks of class. That means every Friday in April and May. Quizzes will be closed book.

Final exam: Monday, June 5, 8:00-10:00am.

The final exam is closed book, with the exception that you may consult two pieces of paper during the exam. You may write or print whatever you wish on each side of these pieces of paper.

Emergencies: If you have to miss an assignment deadline or a quiz because of a medical or other emergency notify me as soon as possible and we will negotiate a substitute. If you know in advance that you have to miss a deadline, let me know at least a week in advance.

Grading: Students will be graded on the three programming assignments, the eight quizzes, some in-class pop quizzes (which are unscheduled and which we will solve together in class), and the final exam.

Grades will be assigned based on scores as shown. At the discretion of the instructor, scores may be scaled. Awarding ± is at the discretion of the instructor.

Program 1	Program 2	Program 3	Quizzes	Pop quizzes	Final
10%	20%	20%	20%	5%	25%

%	90-100	80-89	70-79	60-69	0-60
Grade	A	B	C	D	F

Academic dishonesty: Please read Appendix D of WWU's Catalog on Academic Dishonesty. It is available online at `http://catalog.wwu.edu`.

Unless specified otherwise, all work for this course is meant to be done **individually.** The work that you turn in for a grade must be completely your own, or you will be guilty of academic dishonesty and could receive an F for the course.

However, it can be a valiable learning experience to discuss work with your fellow students, and this is encouraged. However, after working with a colleague, **you may not keep any paper or electronic copies of anything you produced together!** You may only keep your memories. In particular, this means that **you may not ask for or give help while sitting in front of a computer where the assignment is open!** Also, **you may not use anything a colleague has emailed to you!** Delete the email and do not save a copy.

To help understand what I mean, remember the **Long Term Memory Rule**. You may discuss, sketch, write things down, use your computers, whatever, but after you are done working with your fellow students all files must be deleted, whiteboards erased, and all papers you created must be destroyed. You should then watch a rerun of *the Simpson's*, play a game of ping-pong, take a walk, or something else for half an hour. After this you can go back to your assignment (alone) and use the knowledge you have now gained.

It is very easy for us to detect copied assignments. Please do not put us in a situation where we have to fail you for plagiarism.

Topics: Topics may change slightly as the course progresses.

- Analysis of algorithms (informal)
- Big-O notation
- Sorting
  - Insertion sort
  - Merge sort
  - Quick sort
  - Radix sort
- Graphs
  - Graph terminology
  - Graph data structures
  - Topological sort
  - Shortest Paths through a DAG algorithm
  - Breadth-first and depth-first traversals
  - Dijkstra's
  - Prim's

  - PageRank
- Trees
  - Tree terminology
  - Tree data structures
  - Expression trees
  - Tree traversals (using expression trees and *-fix notation as a motivating example)
  - Coding trees and huffman trees
  - BSTs
  - AVL trees
  - Splay trees
- Heaps
  - Heap sort
- Hash tables
  - Separate chaining
  - Rehashing
  - Open addressing
  - Hash functions