Western Washington University
Computer Science Department
CS 241
Data Structures
Syllabus, Spring 2015

About this Course

Instructor: David Palzer, Office CF465, Phone 360-650-2249
Email: mailto:David.Palzer@wwu.edu?subject=CS241
Office Hours: ???, or by arrangement
Course Coordinator: Brian Hutchinson, Office CF475, Phone 360-650-4894
Email: mailto:Brian.Hutchinson@wwu.edu?subject=CS241

Class Hours

Lecture (AW203)	MTWF 10:00am—10:50am	

Prerequisites

CS 145 or equivalent; any of Math 125, Math 157

Examples, assignments, and programs will require use of mathematics.

Description

This course introduces students to the design and implementation of some of the most commonly used data structures, including hash tables, graphs and trees. It also introduces several sorting algorithms along and the notion of algorithmic complexity. Programming in Java is required in implementation of concepts.

Course Materials

Course Web Site (Canvas)

The web site for this course is on WWU's Canvas site (https://wwu.instructure.com/). You should be automatically enrolled in the course.

The CSCI 241 page on Canvas will contain a list of announcements intended for the whole class. You are responsible for these announcements.

Email

Some announcements will be sent to your WWU email account. You will be responsible for the content of class announcements made over email. Please ensure that the sender mailto:David.Palzer@wwu.edu is "white listed" on your email account. Check your WWU email daily. Because not everyone has access to email during the weekends, official announcements will not be sent out after 4 pm on Fridays. Likewise do not expect a quick reply to an email sent to your instructor or TA over the weekend.

Also, please put CS241 in the subject line of any email sent to the instructor or TA. This will ensure that the email doesn't fall victim to the dreaded spam filter.

Textbook

The first textbook for the course is *Build Java Programs, a Back to Basics Approach, Third Edition*, Stuart Reges, Marty Stepp, 2014. There is an online site for the text book at http://www.buildingjavaprograms.com/supplements3.shtml.

There is an online version of the textbook available at http://www.coursesmart.com/9780133379440. This version of the textbook is less expensive than the paper version.

The second textbook for the course is *Introduction to Algorithms, Third Edition*, Thomash, Leiserson, Rivest, Stein, 2009.

Course Outcomes

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.

Auxiliary Course Outcomes

- To improve your understanding of how and why to use version control for software development.
- To improve your understanding of how and why to use pair programming.
- To improve your programming skills and your familiarity with Java.

Schedule

The course schedule is available as a web page on canvas. The schedule will be revised as the quarter progresses. Announcements of schedule changes will be posted on canvas.

Week	Dates	Topic	Assignments
1	3/30-4/3	Course Overview, Analysis of Algorithms	
2	4/6-4/10	Big-O Intuition, Big-O Definition, Analysis of Algorithms, Insertion Sort	Prog 1 Assigned
3	4/13-4/17	Merge Sort, Quick Sort, Radix Sort	
4	4/20-4/24	Graphs, Breadth First Traversal, Depth First Traversal	Prog 1 Due, Prog 2 Assigned
5	4/27-5/1	Topological Sort, Prim's Minimum Spanning Tree, Dijkstra's Shortest Paths, Trees	
6	5/4-5/8	Midterm, Tree Representations, Expression Trees, Tree Traversals	
7	5/11-5/15	Binary Search Trees, Tree Sort, Tree Rotations, AVL Intro, AVL Trees	Prog 2 Due, Prog 3 Assigned
8	5/18-5/22	Splay Trees, B Trees, 2-3 Trees, Heaps	
9	5/25-5/29	Heap Sort, Tries, Direct Addressing, Hash Function Basics, Chaining, Rehashing	
10	6/1-6/5	Open Addressing, Hash Functions, Coding Trees, Huffman Codes	Prog 3 Due
Final Exam: Thursday, June 11th, 8:00am-10:00am			

Grading

Grades will be based on the following percentages:

- Exams: 35 %
  - Midterm Exam: 15%
  - Final Exam: 20%
- Quizzes: 20%
- Assignments: 45%
  - Program 1 (pair), 10%
  - Program 2 (individual), 15%
  - Program 3 (pair), 20%

Grades will be assigned on the total of the assessment items according to the following percentages:

Percentage	Grade
90-100	A
80-89	B

70-79	C
60-69	D
< 60	F

The instructor may choose, at his discretion, to curve the grades on individual assessment items (exams, projects, labs) to conform to these percentages. Additional extra credit opportunities are at the discretion of the instructor as well.

The use of '+' or '-' discriminators for the final grade is completely at the discretion of the instructor.

Assignments

There will be three programming assignments. The assignments will be SVN. Programs will be graded on both correctness, does the program run and product the correct output, and design and style.

Exams

The midterm exam is scheduled for Monday, February 9th during class time.

The final exam will be given on Tuesday, March 17th from 8:00am-10:00am.

Attendance, Late Assignments, etc.

If electronic lecture notes are used, they will be available on-line. In general, if you miss a lecture, please review the lecture notes on Canvas and the relevant material in the text book. If you still have questions after reviewing the lecture notes, I am more than willing to discuss the material, answer questions, etc.

Sometimes life brings about unexpected events in the form of a medical or family emergency. If something like this should happen to you during the quarter and it causes you to miss an exam or assignment due date contact the instructor immediately. **No late work will be accepted**. Early, late, or make-up exams will be administered only in extreme circumstances. Be advised that in order to reschedule or make up any missed exam you must provide a valid documented reason: family emergency, illness, military service, school sponsored athletic events, etc. A note from your mother will not be acceptable! The instructor may, at his option, choose to not give a make-up exam and adjust percentages for the final grade appropriately.

Submitting Your Work

Programs will be developed and stored in Subversion version control repositories, which can be accessed in Windows using the TortoiseSVN client, and in Linux using the command line Subversion client. Fear not - technical details of this process will be explained. Students will be expected to version their programs from start to finish. Be

sure to check in your final version before the assignment deadline - I will check out the most current version of the repository when the clock strikes due. You should also check in your code before coming to office hours if you would like to receive feedback on it.

Academic Dishonesty

Academic dishonesty is defined in the University Catalog as misrepresentation by deception or by other fraudulent means which compromises an instructor's ability to fairly evaluate a student's work or achievement. It is the instructor's responsibility to confront a student and to take appropriate action if academic dishonesty, in the instructor's judgment, has occurred. Please refer to the University Catalog for further information.

Any student who violates the University policy on academic dishonesty will receive an F grade for the course.

Copying files or other documents from someone else and claiming they are your own is plagiarism. Providing files that you created to another student or being party to such actions also amounts to academic dishonesty.

It is very easy for experienced software developers like your instructor and your teaching assistants to detect copied assignments. Please do not put me in a situation where I have to fail you for plagiarism.