

Computer Science 241: Data Structures

Fritz Ruehr • Willamette University • Spring 2019

Introduction

The study of data structures and algorithms serves as a basic foundation to a Computer Science education. As a second course in programming, it enriches a student's understanding of the basic processes involved in computing ... but it also begins to focus attention on deeper and more abiding issues. In this course, we shift our attention from simple coding techniques to the analysis of algorithms, in terms of resource use (time and space), generic solutions to recurring problems, and larger-scale program design. A good portion of our time will be spent becoming familiar with the discipline's standard repertoire of data structures and algorithms. In order to support larger-scale design, we will stress principles of abstraction and modularity. We will try to divide our programs into cleanly-separated components, with narrow interfaces, and consider the specification of their behavior separate from its possible implementations. Through all of this, our programming vehicle will be the modern, object-oriented programming language Java. Students should come out of this course with a solid capability for programming and design and a good foundation for future study of Computer Science in general.

Instructor

Name: Fritz Ruehr

Office: Ford Hall 208

Phone: 370-6165

E-mail: fruehr@willamette.edu

Course homepage: <http://www.willamette.edu/~fruehr/241>

Office hours: M 4:00-5:00; Th 1:00-2:00; or by appointment!

Computer Science Tea: an informal gathering every **Thursday from 11:30-12:50 in the CS hearth**

You are invited!

Logistics and attendance

Classroom time will be spent mainly in lecture; however, you are strongly encouraged to attend lab to work on your assignments when possible, or to meet with the instructor at some other time if you are having trouble and cannot make it to lab.

Lectures MWF 10:20-11:20 Ford 202

Lab times MWF 11:30-12:30 Ford 202

You are responsible for all content, changes in assignments or policies, etc., announced during the course of scheduled lectures; if you are unavoidably absent, inquire at the next class meeting, in office hours or by email.

Grading of programming projects will be based on an interactive, hands-on “demo:” when you are confident that your program is complete and correct, you will seek me out (preferably during lab) and ask to show me your work. I will look over your running application and your written program, try different inputs and features, verify that your program is correct and well-designed, and giving you feedback as appropriate. If I find errors or problems that could be easily fixed, I will try to give you an opportunity to make changes before the due date.

We feel that this is a flexible and humane approach to the grading process which maximizes interaction between students and faculty.

Textbook

We will use **Data Structures and Algorithm Analysis in Java** (3rd edition), by Mark Allen Weiss; it may not be available in the bookstore—see the first day of class for details. ISBN: 0-13-257-627-9.

Language and tools

We will use the Java programming language and the NetBeans IDE (see netbeans.org); it should be installed on all lab machines and is a free download.

Grading policy

Grades will be based on programming projects, exams and class participation—in-class quizzes and written homework may also be included, as needed to determine student progress. Individual grades will be given in numeric form and then combined to determine an overall grade at the end of the semester, according to the weights listed below.

Much of the time you devote to class will be spent writing programs. I plan to give one shorter mid-term and a longer, comprehensive final (during the regularly scheduled exam period—see below). The weights used for the overall grade will be:

- 45% divided evenly among the programming projects;
- 10% total for in-class quizzes, other class participation, and any written homework;
- 20% for the mid-term exam; and
- 25% for the final exam.

(And see above regarding the “demo” style of program grading.)

Programming projects will generally be due about one week (at the beginning of the course) or two weeks (toward the end of the course) after they are assigned. On occasion, class-wide extensions may be announced for various reasons. If you think you will be unlikely to be able to complete an assignment, contact me **before** the due date regarding the problem.

Note: the **final exam** is scheduled (by the College) for **Monday, May 13, 2019, from 8-11 am;** ***please make your summer travel plans accordingly!***

Collaboration and related issues

All programming projects, exams and other work you hand in should be your own. You are allowed (and encouraged) to seek help from other students for general study purposes, but you should never allow other people to do your work *for* you. You may use standard libraries supplied with NetBeans, but you should leave all copyrights and attributions intact and clearly identify your own contributions. Violations of these rules will result in penalties according to College policies, but could include a zero grade for the assignment or a failing grade for the course. In essence, it is OK to discuss ideas with other students, but to copy a piece of writing or program from another source (unattributed), or to have someone else dictate it verbatim, write it down, or type it in for you is NOT allowed.

Topical coverage

We will follow the Weiss textbook fairly closely, although we will not likely make it through the whole book (a later course, *CS 343 Analysis of Algorithms*, covers the remaining topics (and more). Where time permits, I will supplement the text in a few places with outside material (e.g., Chris Okasaki's RA-lists).

Topics we cover this semester should include:

- course introduction and basic concepts
- Java review and new features (exceptions, interfaces, generics, inner classes)
- asymptotic analysis and O-notation
- recursion and induction
- linked lists and array-lists
- stacks and queues
- trees, including algebraic expression trees
- binary search trees, AVL trees, splay trees (and possibly B-trees)
- hash tables, including open and closed hashing and probing strategies
- priority queues and heaps
- sorting algorithms

College-wide policies

Accreditation information

In accordance with college accreditation initiatives, the following are the **student learning objectives** associated with this class (SLO numbers are relative to CS Department norms):

CS/SLO #1: “Students will achieve proficiency in discrete math.”

CS/SLO #2: “Students will achieve proficiency in Computer Science skills (fundamentals of programming, computer organization, architecture, algorithms, theory, designing and implementing software).”

CS/SLO #5: “Students will demonstrate the ability to work independently to analyze and solve problems.”

Time commitment for this course

“Willamette's Credit Hour Policy holds that for every hour of class time there is an expectation of 2-3 hours work outside of class. Thus, for a class meeting three days a week you should anticipate spending 6-9 hours outside of class engaged in course-related activities. Examples include study time, reading and homework assignments, research projects, and group work.”

ADA compliance

“Please tell me about any disabilities that will affect your participation in this course. I will respect any accommodations authorized by the Office of Disabilities Services.”

Plagiarism and cheating

For full details of the College policies and procedures regarding plagiarism and cheating, see:

http://www.willamette.edu/cla/catalog/resources/policies/categories/plagiarism_cheating.php

Important dates

For the official college academic calendar for this year, see:

<http://www.willamette.edu/dept/registrar/calendar/cla/index.html>

Commitment to Positive Sexual Ethics

Willamette is a community committed to fostering safe, productive learning environments, and we value ethical sexual behaviors and standards. Title IX and our school policy prohibit discrimination on the basis of sex, which regards sexual misconduct — including discrimination, harassment, domestic and dating violence, sexual assault, and stalking. We understand that sexual violence can undermine students’ academic success, and we encourage affected students to talk to someone about their experiences and get the support they need. Please be aware that as a mandatory reporter I am required to report any instances you disclose to Willamette's Title IX Coordinator. If you would rather share information with a confidential employee who does not have this responsibility, please contact our confidential advocate at confidential-advocate@willamette.edu. Confidential support also can be found with SARAs and at the GRAC (503-851-4245); and at WUTalk - a 24-hour telephone crisis counseling support line (503-375-5353). If you are in immediate danger, please call campus safety at 503-370-6911.