

## Syllabus

CSE373: Data Structures and Algorithms for Nonmajors

Instructor	Gary Yngve
Email	<a href="mailto:gyngve@cs.washington.edu">gyngve@cs.washington.edu</a>
Phone	206-221-6311 (email preferred)
Office	CSE 370
Office hours	Tuesdays 11:30-1:30 or by appointment

### 1 Textbook

Goodrich and Tamassia, *Data Structures and Algorithms in Java*, 4th ed. (There are substantial changes from the 3rd edition, including the adoption of Java 1.5.)

### 2 Course Overview

Fundamental algorithms and data structures. Theoretical analysis, implementation, and application. Java Collections Framework. Lists, stacks, queues, heaps, hash maps, balanced trees, sets, graphs. Searching and sorting.

Prerequisite: CSE143

This course is heart of computer science. Having gone through the introductory courses, you now have enough background to study the data structures and algorithms that are omnipresent in software. We will study them across a wide spectrum, ranging from theoretical arguments about correctness and efficiency to the nuts and bolts of the implementation to high-level design choices of when to choose one instead of the other in an application.

Upon completing this course, you will have a good understanding of the common data structures and algorithms in computer science, you will have experience using them in code, and you will be able to assess the pros and cons of using them to do a given task.

### 3 Course Webpage

The course webpage will be continually updated over the course of the quarter, with information on assignments, handouts, etc. It includes links to the course messageboard and announcement archive as well as other useful information.

<http://www.cs.washington.edu/373>

### 4 Exams

This course has two midterms and a final. All exams are closed-book, closed-notes, no calculator. See the calendar on the webpage for exam dates.

There are no makeups allowed for exams. At my discretion, I may grant exceptions given extenuating circumstances. Unless humanly impossible, you must notify me prior to the exam (even if you're sick in bed at home, you should still be able to make a phonecall).

## 5 Assignments

This course has four full assignments plus a warmup assignment. Each assignment has both programming components and short-answer components. The short-answer components may ask you to analyze your code's efficiency or your design choices. Or they may be more like book problems, independent of the programming. The short-answer components are turned in electronically, as a txt file. Some assignments may have bonus – it is intended to challenge people and let them express their creativity. The bonus will probably require a lot of time for just a few points.

I strongly encourage you to start early on the assignments so there's plenty of time to get help if you get stuck on some nasty bugs. I've been programming for twenty years, and I'm still no good at predicting how long it will take me to code something. (Managers joke that they ask their developers how long it will take and multiply by three.)

### 5.1 Late Policy

All assignments are due electronically by 10 PM on turnin server linked from the webpage. Refer to the course calendar for due dates. Each student in the class will have a total of four “free” late days (a late day is 24 hours of lateness). There are no partial days, so assignments are either on time, 1 day late, 2 days late, etc. Once a student has used up all his or her late days, each successive late day will result in a loss of 25% on that assignment.

All assignments must be submitted by 10 PM, Sunday Dec 11, whether or not a student has free late days left.

### 5.2 Policy on Collaboration

You are to complete assignments individually. You may discuss the assignment in general terms, but the code you write must be your own. You are encouraged to discuss ideas, approaches, concepts, bugs, etc., in English, but you may not show or give your code to anyone except this course's TAs and instructor. You are not allowed to write code with another student on an assignment or to show another student your solution to an assignment.

The course messageboard is a good medium for getting help on assignments (please do not post your code for an assignment on the messageboard though – if you are unsure if something is acceptable to post, email us). You can also email the instructor or TAs or go to office hours. Each TA will hold two office hours per week, the times to be determined and posted on the webpage.

## 6 Grades

My goal is for the median grade of the course to be a 3.2, corresponding to an average of 80%. Exams and assignments are partitioned as follows:

<b>Exams</b>	<b>50%</b>
first midterm	16%
second midterm	16%
final	18%
<b>Assignments</b>	<b>50%</b>
hw0	6%
hw1	11%
hw2	11%
hw3	11%
hw4	11%

## 7 Computing Resources

We strongly recommend that you use Java 1.5 for the assignments. You must use a minimum of 1.4.2. Though we will not be using generics on homeworks, the textbook uses them. Good text editors include Textpad, Dr. Java, and Eclipse. See handout #2 for instructions on how to install the software on your home machine. The Math Sciences Computing Center is the designated lab for this course; they have the above software installed.

<http://www.ms.washington.edu/mscc/lab.html>

Some minutae on the lab computers: Students need to use the Temp folder linked on the Desktop. Their contents will be deleted after logging out, so the students are responsible for saving their code on Dante or somewhere else via SSH, USB card, etc. The default behavior for clicking on a Java file is Visual J++, so right-click to open with TextPad or whatever else you want.