

CSE 373 – Data Structures and Algorithms

Instructor: Kasey Champion

Email: champk@cs.washington.edu

Office Hours: Fridays 2:30-5:30pm or by appointment (evenings or mornings only)

Office: CSE 216

Please contact cse373@cs.washington.edu with questions about registration or administrative concerns

Times:

- Lecture: KNE 110 from 8:30 to 9:20, MWF
- Midterm: Friday April 27th in class
- Final: Tuesday, June 5th KNE 110 8:30 to 10:20

Course Objectives

The primary focus of this class is on gaining a strong theoretical and conceptual understanding of common data structures and algorithms as well as how they can be practically applied within larger programming projects.

Specific topics we will cover include:

- **Data structures and ADTs:** lists, dictionaries, trees, balanced trees, AVL trees, hash tables, priority queues, binary heaps, disjoint sets
- **Graphs and graph algorithms:** graph search, shortest path, and minimum spanning trees
- **Algorithm analysis:** asymptotic analysis, locality, P and NP complexity classes
- **Sorting and divide-and-conquer**

This course is also designed to have a practical component to help you gain basic familiarity with techniques used within industry. In particular, you'll be asked to:

- Work on larger projects in pairs
- Learn how to use an industrial-strength IDE
- Learn techniques for checking correctness (writing unit tests with JUnit, designing and checking invariants, etc)
- Gain basic exposure with integrating and using 3rd party libraries

And finally, this course also emphasizes the importance of making and justifying effective design decision. All of the above skills are those that are traditionally a part of any technical interview, thus this course is typically considered an important pre-requisite when applying for development positions.

Grade Breakdown

Item	Weight
Individual Written Assignments	15%
Partner Projects	50%
Midterm	15%
Final	20%

Homework

There are two types of homework assignments: solo written assignments and partner programming projects. Homework due dates will be Fridays before 11:59pm to the Canvas turn in system.

Individual Written Assignments

There will be 3 written assignments throughout the quarter, each span 1 week in length. These must be completed INDIVIDUALLY. Please see section on academic honesty and collaboration for more details.

Programming projects

There will be 4 pair programming projects throughout the quarter. These are meant to be completed in pairs, though you may work on them individually if you so choose. Projects generally span multiple weeks with a checkpoint due once a week. These checkpoints are due at the normal Friday night time.

If you lose points during a checkpoint, you may earn back half the points you lost if you fix and resubmit your work for the next project deadline. For example, suppose you earned 60% for your project 1, part 1 submission. If you fix and submit your part 1 code by the part 2 submission deadline, your part 1 score will be retroactively adjusted to 80%.

Note that your final project submission will not be eligible for this regrade policy.

Grading

Grading will be done by the TAs.

Grading policies for written assignments: Your written work will be graded based on correctness as well as the quality of the justifications and explanations you provide.

Grading policies for partner projects: Your code will be graded based on the following criteria:

1. **Correctness:** Does your code pass the tests we gave you? For every project we will withhold some extra tests, so be sure to add extra tests of your own. Full passes on given tests does not guarantee full credit for correctness.
2. **Design:** Are there any larger design issues with your code (e.g. inefficiency or gross redundancy)?
3. **Style:** Does the automatic style checker we've provided you report no issues with your code?
4. **Test quality:** You will sometimes be asked to write tests for your data structures. Do these tests correctly identify problems in buggy implementations?

If you feel you incorrectly lost points on an assignment, please start by talking to your grader to try and resolve the issue with them. If the issue is still not resolved, fill out the regrade request form linked on the class website. All regrade requests must be made within a week after you receive your feedback.

Late Days

Each student receives 3 late days that may be used for written assignments, checkpoints, and project due dates. If a team submits a project late, both students must use a late day. If you (or your partner) have no late days left, you will be deducted 20% of your overall score per each day late.

We will not accept assignments two days after the deadline, whether you use late days or not. Late days will be applied automatically. Leftover late days will have zero impact on your grade.

If unusual circumstances truly beyond your control prevent you from submitting an assignment or attending an exam on time, you should discuss this with the course staff ASAP. If you contact us well in advance of the deadline, we may be able to show more flexibility in some cases.

Extra Credit

We will offer extra credit from time to time. Completing extra credit will **NOT** have a major impact on your grade: we will keep track of extra credit independently of your homework and use them to bump up borderline grades at the end of the quarter. This means completing extra credit will not be an effective way of making up for missed/incomplete assignments or boosting your grade.

Rather, the purpose of extra credit is to give you opportunities to engage more deeply in material you personally find interesting or to try implementing more challenging material. They're meant to be fun!

Exams

There will be 2 timed exams in this course: a midterm and a final exam. These will be written exams and you will be allowed a single sheet of 8.5"x11" paper for notes.

Make-up exams will only be given in case of a serious emergency. If you miss an exam, even if you are sick or injured, you must contact Kasey BEFORE the exam (or arrange for someone to do so). You must show evidence that you are physically unable to take the exam. No make-ups will be granted for personal reasons such as travel or conflicting schedules. No accommodations will be made for students who arrive late to exams. The only accommodations for exams that will be made are those that correspond to the University's official accessibility guidelines, which must be reflected in your student account.

Software and Textbooks

We will use Java 8 on all programming assignments. (Our assignments should also theoretically be compatible with Java 9, but Java 9 is very new, so use it at your own risk).

Our officially supported IDE is Eclipse. You may use other IDEs like IntelliJ or Netbeans if you wish, though the course staff may not necessarily be able to help you with any setup-related questions you may have.

However, you **MAY NOT USE jGrasp**. (Well, you can try, but it's very likely that jGrasp will not be able to correctly load or run our class projects.)

The textbook for this course is *Data Structures and Algorithm Analysis in Java (3rd edition)* by Mark A. Weiss. This textbook is **OPTIONAL**: you should not need it to succeed in this course.

Academic honesty and collaboration

Any assignments that has copied or paraphrased content will receive a zero. You are still, however, encouraged to discuss assignments and concepts on a high level with your classmates so long as you:

1. Make a serious attempt on each and every problem or coding project before discussing it with others.

2. Limit cooperation to just group discussion and brainstorming. A good rule of thumb is that you should not take away any written notes, photographs, or other records from your brainstorming session.
 - a. Do not show your code to another student, nor should you look at another student's code directly. This includes current and former students, tutors, friends, paid consultants, people on the internet etc...
 - b. Do not post your homework solution code publicly online for any reason.
3. Write up each and every problem in your own writing, using your own words. (Similarly, you must write code on your own).

Naturally, when working on a partner projects, you're free to share your code and collaborate in full detail with your partner. However, you should not share your code with other groups.

If you have a specific question about your code or written work, the only person you may show your work is a member of the course staff.

If you do not follow these rules, you will be considered to have cheated. Cheating is a very serious offense. If you are caught cheating, you can expect a failing grade and initiation of a cheating case in the University system. Please note that the above consequences apply to those that provide too much assistance, not just the "copier". If you're not sure if something constitutes cheating, send the instructor an email describing the situation.