CSE 373 18wi: Syllabus

Overview

Teaching staff:

- **Instructor**: Michael Lee (mlee42@cs.washington.edu)
- **Co-instructor/supervisor**: Adam Blank (blank@cs.washington.edu)
- **Course staff and office hours**: See course website (https://courses.cs.washington.edu/courses/cse373/18wi/)

Times:

- **Lecture**: Gowen 301 from 3:30 to 4:20, MWF
- **Midterm**: Friday, Feb 2. Time and location TBD
- **Final**: Thursday, Mar 15 from 2:30 - 4:20, GWN 301

Goals

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, AVL trees, hash tables, priority queues, binary heaps
- **Graphs and graph algorithms**: graph search, shortest path, and minimum spanning trees
- **Algorithm analysis**: asymptotic analysis, locality, P and NP complexity classes
- **Sorting, divide-and-conquer, algorithm design**

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)
- Gain basic exposure with integrating and using 3rd party libraries

And finally, this course also emphasizes the importance of making and justifying effective *design decisions*.

Homework

There are two types of homework assignments: solo written assignments and partner programming projects.

Written assignments span about one week each and MUST be completed INDIVIDUALLY. See the section on academic honest and collaboration below.

Programming projects are meant to be completed in pairs, though you may work on them individually with permission. Projects generally span multiple weeks with a checkpoint due once a week.

If you lose points during a checkpoint, you may earn up half the points you lost back 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. So, if your part 2 submission earned a 75%, that will be your final grade for part 2 (assuming the project consists of two parts).

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.

Important: the projects are designed to be cumulative so it is paramount that you do not fall behind. If you are struggling to complete a particular assignment, you should get help ASAP – see the related section below.

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.

Academic honesty and collaboration

We take academic honesty very seriously. Cheating not only deprives you of practice and understanding, but is also unfair to the rest of your class. 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:

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

(b) 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.

(c) Write up each and every problem in your own writing, using your own words. (Similarly, you must write code on your own).

(d) List each person you collaborated with at the top of your written homework or in your project writeup.

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.

Cheating: 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. Cheating is an insult to other students, to the instructor, and to yourself. If you feel that you are having a problem with the material, or do not have time to finish an assignment, or have any number of other reasons to cheat, we have many mechanisms for getting help. Copying the work of others is not the solution.

For more details, see this academic misconduct page linked on the course website. If you're not sure if something constitutes cheating, send the instructor an email describing the situation.

Getting help

There are two main ways of getting help:

- **Piazza**:

  We use Piazza as an online Q&A forum where you may ask questions (both publically and privately). If you have a question, this should be the first place to check – it's often the case that other students may have had the same question.

  We also strongly recommend you monitor new questions asked on Piazza semi-regularly: seeing questions other students ask can often be surprisingly enlightening.

- **Office hours:**

  Please visit office hours! Answering questions during office hours is often one of the best parts of teaching, so please do us a favor and make use of them.

One additional resource to keep in mind is the resources page on the class website. This page contains links to a variety of guides, tips, and tricks on material we don't have time to fully cover during lecture or section. The resources there might not always be relevant, but they're checking from time to time.

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!

If you don't find the offered extra credit assignments compelling, you may also design your own extra credit assignments, with approval. If you plan on taking this approach, you should schedule a meeting or visit office hours to make sure your proposed extra credit is on-topic and appropriately scoped.

Assessment

Here is how your grade will be calculated:

- Written assignments: 15%
- Partner projects: 45%
- Midterm Exam: 20%
- Final Exam: 20%

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:

- **Correctness**: Does your code pass the tests we gave you? Does it pass some additional tests we've withheld?
- **Design**: Are there any larger design issues with your code (e.g. inefficiency or gross redundancy)?
- **Style**: Does the automatic style checker we've provided you report no issues with your code?
- **Test quality**: You will sometimes be asked to write tests for your data structures. Do these tests correctly identify problems in buggy implementations?

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.

Acknowledgements

Many of the materials posted here and used in this course have been shared and refined by many other instructors and TAs in previous offerings of CSE 373, CSE 332, and CSE 326. This version of the course was particularly influenced by previous offerings by Riley Porter, Ruth Anderson, Dan Grossman, Kevin Quinn, Hunter Zahn, Evan McCarty, and Adam Blank.