CS 271: Data Structures

Term: Fall 2022
Credits: 4
Lecture: 1:50 PM – 2:50 PM, MTF in HS 309
Lab: 1:50 PM – 2:50 PM, Th in HS 101C
Prerequisites: CS 262 with a grade of C or better.

Instructor

Instructor: Scott Summers
Email: summerss@uwosh.edu
Phone: 920-424-1324
Office: Halsey 220
Office hours: Indicated by the cells shaded in gray in the following table and subject to change.

	Mon	Tues	Wed	Thu	Fri
9:00 AM					
9:30					
10:00					
10:30					
11:00					
11:30					
12:00 PM					
12:30					
1:00					
1:30					
2:00					
2:30					
3:00					

As indicated above, I hold six total scheduled office hours per week. You don't need to schedule an appointment with me if you want to stop by during my scheduled office hours. If you want to meet virtually, via Microsoft Teams, then send me an email a few minutes before you'd like to meet, simply asking if I'm available to meet virtually. I'll respond as soon as I'm available. If you'd like to meet with me outside of my scheduled office hours, then please email me and we'll work out a time that works for both of us. If my office hours ever change, even for a single day, then I'll send the class updates via the course webpage.

Course description

A course surveying the fundamental methods of representing data and the algorithms that implement and use those data representation techniques. Data structures and algorithms include; linked lists, stacks, queues, trees, heaps, priority queues, hashing, searching, sorting, data compression, graphs, recursion. Analysis topics include: elementary big-O analysis, empirical measurements of performance, time/space trade-offs, and identifying differences among best, average, and worst case behaviors.

Learning outcomes

At conclusion of the course, students will be able to:

1. Given a non-recursive algorithm, the student will be able to examine its loop structures and infer its asymptotic runtime using big-O notation.
2. Given a recursive algorithm, the student will be able to examine its recursive structure, determine the corresponding recurrence relation (from a small collection of commonly occurring recurrence relations), and use the recurrence relation in determining the asymptotic runtime of the algorithm using big-O notation.
3. Given the description of a computational problem requiring a mixture of search, insertion, and/or deletion operations on collections of data, the student will be able to compare the relative advantages of using arrays and linked lists in solving the problem efficiently.
4. Given a classical computational problem (e.g., infix-to-postfix conversion, postfix-expression evaluation, path planning, minimum-spanning tree computation), the student will be able to trace a solution to the problem using appropriate data structures (e.g., stacks, queues, binary trees, binary search trees, red-black trees, graphs) and to predict the asymptotic runtime of the solution based on the selected data structures.
5. Given a collection of unordered data, the student will be able to trace the execution of an advanced sorting algorithm (such as quick sort and heap sort) on this data set.
6. Given a set of data keys, the student will be able to trace through a sequence of key insertions, searches and deletions on a balanced tree structure. The student will also be able to discuss the relationship between the number of keys and the execution time of these operations.
7. Given a set of data keys, a hash function, a table size, and a collision-handling strategy, the student will be able to trace through a sequence of key insertions and searches, and to discuss how varying the table size, hash function or collision-handling would affect the execution time of these operations.
8. Given a graph data structure, the student will be able to implement it using either adjacency lists or an adjacency matrix, to traverse it using either a depth-first or breadth-first strategy, to identify its structural properties (whether it is directed, cyclic, connected, complete), and to trace the execution of one or more classical graph algorithms (e.g., Dijkstra's shortest path, topological sort or minimum-spanning tree computation).
9. Given a problem requiring the efficient use of a variety of data structures, the student will be able to apply object-oriented design principles in implementing and testing a solution to that problem in an appropriate object-oriented language.

Course website

The course website is: `https://uwosh.edu/canvas/`. You should check Canvas on a regular basis, perhaps two or three times per day.

There is no required textbook

There is no required textbook for this course. I will present in lecture and lab all of the content that you need to know to be successful in this course.

Course grade

Your final course grade will be based on the following types of assessment items.

10% QUIZZES

You will be given one quiz roughly every week. Quizzes will be taken at the end of class. Each quiz is equally-weighted. Calculators will not be allowed for any of the quizzes. Each quiz will test your knowledge of the material from roughly the prior week but older material may be covered depending on how the class performs on the prior quiz.

30% LABS

There will be weekly labs. All labs are equally-weighted. In each lab, you will have to write a short-to-medium length Java program comprised of one or more classes.

60% EXAMS

There will be three equally-weighted in-class exams. Exam material will come from the lecture content, quizzes, and labs. There will be more information about each exam as it approaches. All exams will be taken during the regular class period.

Grading scale

Grading will be on a plus/minus system. Grading may be done on a curve depending on the overall performance of the class. If no curve is used, then your grade will be computed based on the following:

Percentage	Grade
> 91	A
> 89 and ≤ 91	A-
> 87 and ≤ 89	B+
> 81 and ≤ 87	B
> 79 and ≤ 81	B-
> 77 and ≤ 79	C+
> 71 and ≤ 77	C
> 69 and ≤ 71	C-
> 67 and ≤ 69	D+
> 61 and ≤ 67	D
> 55 and ≤ 61	D-
≤ 55	F

Re-grading

If you believe an assessment item (e.g., exam, etc.) was graded incorrectly or unfairly and would like to have it re-graded, please let me know about it in writing within one day of receiving the assessment item back. I will re-grade the entire assessment item and you may gain or lose points accordingly.

Late work

If you are unable to take a scheduled exam at its normal time, then ,it may be possible to take a make-up exam provided that you do both of the following, which are then subject to my approval:

1. Make arrangements prior to the scheduled exam (for last minute emergencies, call me at 920-424-1324 or leave a message at the Computer Science office, 920-424-2068. **No after-the-fact notifications will be accepted**.

2. Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

In some cases, only one make-up exam will be given. It will be a comprehensive exam given at an arranged time during the last week of the semester.

Late work will NOT be accepted. Late work is worth 0 points. Extensions may be granted at the discretion of the instructor if you provide a valid justification.

In general, the following are the only valid excuses for not completing or submitting an assessment item on time:

- Medical illness – provide to me an official note from a medical provider (or the Dean of Students) supporting your claim of a medical illness and describing its severity.
- Family death – provide to me the name of the deceased, the name of the funeral home, and your parent's address, to which I can send my condolences.
- Religious holiday – provide to me before the third week of the semester a list of all the dates of conflict.
- University-sanctioned events, e.g., athletic competition, student organization event, conference attendance, career fair, etc. – provide to me before the third week of the semester a list of all the dates of conflict, along with corresponding supporting documentation on official university letter head.

Work is not a valid excuse to miss class. Your job must work around school, and not the other way around.

If you have four or more unexcused absences, then I reserve the right to automatically fail you.

If you have six or more medical absences, then you must medically withdraw from the course.

Coding standards

In this class, you will write several short to medium-length Java programs. One of your goals (during this class and beyond, in Java or any programming language) should be to write understandable, readable code. You should be making every effort to document anything that might be confusing to a reader unfamiliar with your program using correct spelling and grammar, to name variables intelligently, to use indentation that reflects the code's organization, and so on. All of this will be taken into account during grading: poorly organized or written code may have a negative impact on your grade, even if the resulting program works fine. Briefly, your code must be formatted in a consistent and easily-readable manner. At a minimum, I will require that you utilize the auto-format feature of whatever development environment we use (e.g., "Auto-layout" in BlueJ).

One of the goals of this class is to teach you to write functioning programs in Java. Thus, your code must compile and run correctly in order for you to receive full credit. Code that does not compile will receive substantially less than full credit. Keep this in mind when writing programs: write your code in small pieces, making sure each piece works before moving on to the next one. It is much better to turn in a project that is not finished but has many working pieces than to turn in one that doesn't work at all, even though most of the code is written.

University policy on academic integrity

The University of Wisconsin Oshkosh is committed to a standard of academic integrity for all students. The system guidelines state:"Students are responsible for the honest completion and representation of their work, for the appropriate citation of source, and for respect of others' academic endeavors" (UWS 14.01, Wisconsin Administrative Code).

Often, students are not aware of the ways to identify and avoid plagiarism. Therefore, it is important to educate yourself about how to give proper credit to sources that you use in your assignments. For writing

assignments, you can consult the Purdue Owl website on how to identify and avoid plagiarism: https://owl.english.purdue.edu/owl/resource/589/02/ and https://owl.english.purdue.edu/owl/resource/589/03/. This website outlines the strategies for avoiding plagiarism in this course. However, other courses may demand knowing other ways to identify and avoid plagiarism. Therefore, I encourage you to consult with me if it is unclear to you how you give proper credit to your sources of information.

According to the Dean of Students
(see https://uwosh.edu/deanofstudents/student-conduct/academic-misconduct/), examples of academic misconduct include, but are not limited to:

- plagiarism (turning in work of another person and not giving them credit),
- stealing an exam or course materials,
- copying another student's homework, paper, or exam,
- cheating on an exam (copying from another student, turning in an exam for re-grading after making changes, working on an exam after the designated time allowance),
- providing solutions on online forums (e.g., Discord, etc.), and
- falsifying academic documents.

In sum, all material turned in for this course must be original. In this course, you may not re-use papers or projects from other sections of this course, from other courses you have completed, or other courses you are currently completing. This class is a specific event in your learning process. To learn, you must engage in the material and complete the work. Thus, work from other experiences is not acceptable. All work turned in that is plagiarized will receive a "0" in the course.

Accommodations

UW Oshkosh supports the right of all enrolled students to a full and equal educational opportunity. It is the University's policy to provide reasonable accommodations to students who have documented disabilities that may affect their ability to participate in course activities or to meet course requirements.

Students are expected to inform instructors of the need for accommodations as soon as possible by presenting an Accommodation Plan from either the Accessibility Center, Project Success, or both. Reasonable accommodations for students with disabilities is a shared instructor and student responsibility.

The Accessibility Center is part of the Dean of Students Office and is located in 125 Dempsey Hall. For more information, email accessibilitycenter@uwosh.edu, call 920-424-3100, or visit the Accessibility Center's website at https://uwosh.edu/deanofstudents/accessibility-center/.

Statement regarding diversity, equity & inclusion

Diversity drives innovation, creativity, and progress. At the University of Wisconsin Oshkosh, the culture, identities, life experiences, unique abilities, and talents of every individual contribute to the foundation of our success. Creating and maintaining an inclusive and equitable environment is of paramount importance to us. This pursuit prepares all of us to be global citizens who will contribute to the betterment of the world. We are committed to a university culture that provides everyone with the opportunity to thrive.

Required disclosure statement

Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990:
https://uwosh.edu/financialaid/consumer-information/

Tentative daily schedule

The following is a tentative daily schedule (subject to change):

Date	Topic
Thursday, 9-8	Day 1 (Lab1): Attendance, syllabus, Eclipse, using `List`s.
Friday, 9-9	Day 2: Algorithms, data structures, computational complexity, running time complexity, space complexity, simple operations.
Monday, 9-12	Day 3: Worst-case running time, upper bounding the number of simple operations executed. in the worst case, big-O notation, finding the best possible big-O.
Tuesday, 9-13	Day 4: More examples of big-O, inferring the big-O of `List` operations through empirical analysis.
Wednesday, 9-14	NO CLASS. Lab1 due at noon.
Thursday, 9-15	Day 5 (Lab2)
Friday, 9-16	Day 6: Recursion as a problem-solving technique **Quiz 1** covering content from Days 2-5 (Days 2, 3, 4, and 5).
Monday, 9-19	Day 7: Running time analysis of recursive methods.
Tuesday, 9-20	Day 8: Solving recurrence relations with iteration
Wednesday, 9-21	NO CLASS. Lab2 due at noon.
Thursday, 9-22	Day 9 (Lab3)
Friday, 9-23	Day 10: Sorting, Insertion sort implementation and traces. **Quiz 2** covering content from Days 6-9.
Monday, 9-26	Day 11: Quick sort implementation and traces.
Tuesday, 9-27	Day 12: Merge sort implementation and traces.
Wednesday, 9-28	NO CLASS. Lab3 due at noon.
Thursday, 9-29	Day 13 (Lab4)
Friday, 9-30	Day 14: The `List` ADT, `ArrayList`, `LinkedList`. **Quiz 3** covering content from Days 10-13.
Monday, 10-3	Day 15: Implementation and analysis of `ArrayList` operations.
Tuesday, 10-4	Day 16: Continue implementation and analysis of `ArrayList` operations.
Wednesday, 10-5	NO CLASS. Lab4 due at noon.
Thursday, 10-6	Day 17 (Lab5)

The tentative daily schedule is continued on the next page...

Continuation of tentative daily schedule:

Friday, 10-7	Day 18: Implementation of `LinkedList` operations using a single-linked list. **Quiz 4** covering content from Days 14-17.
Monday, 10-10	Day 19: Implementation of `LinkedList` operations using a double-linked list.
Tuesday, 10-11	Day 20: Finish implementation of `LinkedList` operations.
Wednesday, 10-12	NO CLASS. Lab5 due at noon.
Thursday, 10-13	Day 21: NO LAB. Review for Exam 1 in HS 101C.
Friday, 10-14	Day 22: **Exam 1** covering content from days 1-21.
Monday, 10-17	Day 23: LIFO stack, FIFO queue
Tuesday, 10-18	Day 24: Evaluating postfix expressions.
Wednesday, 10-19	NO CLASS.
Thursday, 10-20	Day 25 (Lab6)
Friday, 10-21	Day 26: Hashing, hash tables, separate chaining **Quiz 5** covering content from Days 23-25. **Last Day to Drop Without Late Drop Request Form OR Withdraw.**
Monday, 10-24	Day 27: Open addressing, linear probe, quadratic probe.
Tuesday, 10-25	Day 28: Double hashing, rehash
Wednesday, 10-26	NO CLASS. Lab6 due at noon.
Thursday, 10-27	Day 29 (Lab7)
Friday, 10-28	Day 30: Binary trees, binary search trees. **Quiz 6** covering content from Days 26-29.
Monday, 10-31	Day 31: Binary search tree operations: search and add.
Tuesday, 11-1	Day 32: Deleting an element from a binary search tree, tree traversal methods.
Wednesday, 11-2	NO CLASS. Lab7 due at noon.
Thursday, 11-3	Day 33 (Lab8)
Friday, 11-4	Day 34: Balanced binary search tree, AVL trees, properties of AVL trees. **Quiz 7** covering content from Days 30-33.
Monday, 11-7	Day 35: Adding an element to an AVL tree.
Tuesday, 11-8	Day 36: Deleting an element from an AVL tree, lazy deletion.
Wednesday, 11-9	NO CLASS. Lab8 due at noon.
Thursday, 11-10	Day 37: NO LAB. Review for Exam 2 in HS 101C.

The tentative daily schedule is continued on the next page...

Continuation of tentative daily schedule:

Friday, 11-11	Day 38: **Exam 2** covering content of all the days since the last exam.
Monday, 11-14	Day 39: Min/max priority queue, min/max binary heap, add to a binary heap
Tuesday, 11-15	Day 40: Delete the min element from a min binary heap, re-heapify down, building a min binary heap in linear time.
Wednesday, 11-16	NO CLASS.
Thursday, 11-17	Day 41 (Lab9)
Friday, 11-18	Day 42: Basic definitions from graph theory. **Quiz 8** covering content from Days 39-41.
Monday, 11-21	Day 43: Adjacency matrix, adjacency list, tradeoffs, applications.
Tuesday, 11-22	Day 44: Other graph implementation schemes.
Wednesday, 11-23	NO CLASS.
Thursday, 11-24	NO CLASS. Thanksgiving break.
Friday, 11-25	NO CLASS. Thanksgiving break.
Monday, 11-28	Day 45: Depth-first search.
Tuesday, 11-29	Day 46: Breadth-first search.
Wednesday, 11-30	NO CLASS. Lab9 due at noon.
Thursday, 12-1	Day 47 (Lab10)
Friday, 12-2	Day 48: Dijkstra's algorithm, analysis and trace. **Quiz 9** covering content from Days 42-47.
Monday, 12-5	Day 49: Topological sort.
Tuesday, 12-6	Day 50: Minimum spanning tree problem and applications.
Wednesday, 12-7	NO CLASS. Lab10 due at noon.
Thursday, 12-8	Day 51 (Lab11)
Friday, 12-9	Day 52: Prim's algorithm, analysis and trace. **Quiz 10** covering content from Days 48-51.
Monday, 12-12	Day 53: Kruskal's algorithm, analysis and trace.
Tuesday, 12-13	Day 54: Review for Exam 3. Student Opinion Surveys.
Wednesday, 12-14	NO CLASS. Lab11 due at noon.
Thursday, 12-15	Day 55: **Exam 3** covering ALL course content, but focusing on the content of all the days since the last exam.
Friday, 12-16	Day 56: Make-up exams.