CS 150 - Data Structures and Algorithms - Fall 2024

This course continues the development of object oriented approaches to the design and implementation of software systems. Students will learn to analyze problems, algorithms and develop object-oriented solutions to problems. Students will also learn to use multiple data structures and the accompanying algorithms to store, index and retrieve data. Lecture/laboratory.
The student work in this course is in full compliance with the federal definition of a four credit hour course. Please see the Lafayette College Compliance webpage for the full policy and practice statement.

Instructor information

Jon Dahl
Office: RISC 467
Office hours:
Monday: 10:30am–11:30am, Tuesday/Thursday: 10–10:50am, or by appointment
Email: dahlj@lafayette.edu

Basic course information

Meeting times: TuTh 1:15pm–2:30pm (Section 1), 11am-12:15pm (Section 2), 8am–9:15am (Lab Section 1), 2:45pm–4pm (Lab Section 2)
Textbook: zyBooks CS150: Data Structures and Algorithms
Grading:
zyBook reading assignments (15%),
Projects (20%),
Lab assignments (20%),
Lab Exam (0%, must pass to pass course),
Midterm Exam (20%),
Final Exam (25%).
Your submitted programs (for labs, projects, and exams) will be graded based on:
Correctness: Correctness is an objective criterion, which means that the program behaves according to the requirements explained in the assignment in all possible cases.
Good Style: Good style refers to the way the program text looks. This includes commenting and correct indentation. We will discuss different aspects of good programming style as we learn how

to program.
Good Design: Good design is a harder criterion to describe. It affects both the execution of the program and its readability. Characteristics of a good design will be illustrated throughout the course
Code must compile in order to be graded.
All assignments are to be submitted as BlueJ projects and BlueJ is the sole IDE allowed for the Lab Exam.
Midterm: Tuesday, October 1.
Lab exam: Thursday, October 24.
Final exam: TBA
Prerequisite: CS104, CS105, CS106, CS AP Score of at least 4 , or an approved equivalent.

Detailed course information

Meetings: Regular meetings will be for 75 minutes.
This class has morning lecture sections followed by afternoon lab sections. Lecture sections will consist of traditional lecture, discussion, and in-class practice. Lab sections are primarily for working on the assignments in a setting with immediate help available from the instructor or TA. You will learn by doing computer programming; interacting with your professor and other students; and engaging with your professors instruction as well as external media. Since your learning of the material is an integral part of the class process, it is essential that you attend class. Please let me know if you can't make it to class, with prior notice where appropriate. An absence will be excused if it is caused by illness properly reported, by an emergency properly reported, or by participation in a recognized collegiate activity. Proper reporting includes advance notice, whenever possible. An unexcused absence will result in a 0 for missed work. In particular, You must attend your lab period if you have not submitted the current lab for grading. **Failing to attend lab unexcused will result in inability to submit that week's lab (unless the lab has already been submitted).**

Submission for assignments: You must submit your assignments by due time. If your project does not work as described in the assignment, you are required to hand in an additional text document which specifies what does not work. **No late submission is accepted unless you have a Dean's Excuse.** Moodle assignment submission link will be closed promptly at the deadline. Submissions made after the deadline (for example, via email) will not be graded. If you cannot finish the project before the deadline, please make sure to submit what you have before the deadline.

Course Goals:
After successfully completing this course, the student will be able to:

1. Students will demonstrate the ability to analyze, design, apply and use data structures on various problems both in lab and as projects. The data structures include the important collection classes such as array list, linked list, stack, queue, binary search tree, tree map/set, heap, hash-map/-set, and weighted graph.

2. Students will be able to design, implement and evaluate their solutions to problems assigned in lab and in the classroom.

3. Students will demonstrate the ability to analyze, design, apply and use data structures to solve problems.

4. Students will be able to work in teams to analyze problems and design different solutions (based on needs) to problems presented in the classroom. The solutions will require the use of a variety of data structures and accompanying techniques/algorithms.

5. Students will demonstrate the ability to apply current data structures and algorithms to solve problems. In addition, students will use current testing tools and approaches to evaluate and test their solutions.

6. Students will learn how to experimentally and analytically determine the complexity of various algorithms and operations on data structures. The information will be used to determine the appropriate design choice of data structure and algorithms to solve problems.

Course Outcomes: At the conclusion of this course students will be able to do the following.

1. ABET/CAC Outcome 1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

2. ABET/CAC Outcome 2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

3. ABET/CAC Outcome 5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

4. ABET/CAC Outcome 6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

Approximate weekly schedule

1. Introduction and Java basics

2. Inheritance, interfaces, and abstract classes

3. Recursion and analysis of algorithms

4. Basic sorting, comparison, and working with generics

5. Merge sort, collections, and iterators

6. Linked lists

7. Stacks and queues

8. Stack applications (infix and postfix notation)

9. Trees and traversal

10. Binary search trees and AVL trees

11. Priority queues and heaps

12. Maps, hash tables, and dictionaries

13. Graphs

Names/Pronouns: I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records.

Academic honesty: I expect all students in this course to be honest in all of their academic work. Your further participation in this course is a tacit agreement to commit to acting in an honorable fashion in this academic community and signifies your understanding that your failure to comply with this commitment may result in disciplinary action. I am required to report every suspected case of academic dishonesty to the Dean of the College. Penalties for academic dishonesty are often quite severe.

In-class group work is collaborative. You should confer with other students and with the instructor while working on these assignments.

As copied from above, for homework, you are encouraged to discuss concepts with anyone willing to talk to you. However, it is expected that in all cases the final product handed in for evaluation of your work will be your own. You may not copy or paraphrase the work of another, nor should you use another's work as a model for your own.

During examinations and quizzes, you must not confer with other students, look at their papers, or use any unauthorized sources of information. Additionally, you may not allow another student access to your work. It is also dishonest to seek information about an examination from another student before taking a make-up examination.

Students are also not allowed to use advanced automated tools (artificial intelligence or machine learning tools such as ChatGPT or Dall-E 2) on assignments in this course. Each student is expected to complete each assignment without substantive assistance from others, including automated tools. The use of any prohibited AI tools in this class will be regarded as plagiarism. If you are uncertain whether the AI tools you are utilizing comply with the AI policy of this course, I encourage you to discuss it with me without hesitation. Maintaining academic integrity is of utmost importance, and I am here to clarify any concerns you may have regarding the AI tools you plan to use. It is your responsibility to ask questions if you are uncertain about the usage of any generative AI tools in this course.

See the Student Handbook for a complete statement of the College's Policy on Academic Honesty.

Digital repositories have become a common tool for collaboration. These repositories include: GitHub, Google drive, etc. Do not default to public permissions! Make your work private. If you are working with a group, have one group member create a separate repository for the group, with only the group members having access permission.

At Lafayette College, all course materials are proprietary and for class purposes only. This includes posted recordings of lectures, worksheets, discussion prompts, and other course items. Such materials should not be reposted. Online discussions should also remain private and not be shared outside of the course. You must request my permission prior to creating your own recordings of class materials, and any recordings are not to be shared or posted online even if permission is granted to record. If you have any questions about proper usage of course materials feel free to ask me. Also, if you have any concerns with being recorded during the course please let me know.

Academic honesty (code):

What is cheating?:

Sharing code: copying, retyping, looking at, supplying file.

Describing code: Verbal description of code from one person to another.

Helping write code line by line.

Searching web for solutions.

What is not cheating?:

Talking about general concepts or high level design.

Special arrangements: If you need disability-related accommodations in this class, if you have emergency medical information you wish to share with me, or if you need special arrangements in case the building must be evacuated, please inform me immediately. Please see me privately after class or at my office.
Privacy statement: Moodle contains student information that is protected by the Family Educational Right to Privacy Act (FERPA). Disclosure to unauthorized parties violates federal privacy laws. Courses using Moodle will make student information visible to other students in this class. Please remember that this information is protected by these federal privacy laws and must not be shared with anyone outside the class. Questions can be referred to the Registrar's Office.
Weather: As a residential college, Lafayette rarely cancels classes due to snow. Instead, instructors are asked to consider their own ability to safely make it to campus. I will email the class if this becomes an issue for some reason, and we will then conduct class remotely.

Help

Office hours: My office hours are Monday: 10:30am–11:30am, Tuesday/Thursday: 10am–10:50am, or by appointment. Please do not hesitate to take advantage of this time.
You may also email me questions (dahlj@lafayette.edu), although discussing program design via email can be tricky at times. I will try to answer quickly if it seems reasonable to, but sometimes we will just have to schedule time in person.
MSG drop-in: We will have drop-in MSG sessions available for extra help throughout the week. I will update Moodle with the schedule for these sessions once they are finalized.