CISC-5006: Data Structures

Spring 2023 Time: 08:00 PM – 10:00 PM EST, Tuesdays

Class Syllabus

Instructor: Tamer Aly
Slack: Click here to sign up for slack and join the channel
Course Repository: https://github.com/tamerfrombk/CISC-5006
Office Hours: Set up appointment through email/Slack

Description

In this class, we will discuss different ways to lay out program data in memory. We will review many common techniques to do so including, but not limited to: arrays, lists, stacks, queues, and trees. We will study the spatial and temporal complexities of each data organization method. This course builds on principles in discrete mathematics and on fundamental programming practices.

Textbook

There is no required textbook for the class. I will recommend suggested readings as we go on in the course.

Asking questions and getting help

The best way to ask questions is through the Slack CISC-5006 channel as I check it multiple times daily. Please avoid DM'ing me a question on Slack since this prevents other students from potentially learning something. Instead, post the question on the #cisc-5006 Slack channel. If you have a private matter to discuss, then please feel free to email or DM me on Slack.

Coursework

There are main parts of the course that determine your grade. See "Quality of Submissions" section for a breakdown of how grades will be assessed.

- **Homework programming assignments:** There will be **10** programming assignments each worth **5** points for a total of **50** points.
- **Midterm:** There will be a midterm worth **20** points. The midterm will consist of two sections:
    - A programming section worth **10** points.
    - A written section worth **10** points.
- **Final:** There will be a final worth **30** points. The final will consist of two sections:
    - A programming section worth **20** points.
    - A written section worth **10** points.
- **Attendance:** Attendance is highly encouraged. Attendance and participation in class help give me some leeway when evaluating the grading policy below.

Grading Policy

The grade might be curved since I expect the average to be around B. The final grade is based on the university guideline: A: [93%, 100%], A-: [90%, 93%), B+: [87%, 90%), B: [83%, 87%), B-: [80%, 83%), C+: [77%, 80%), C: [73%, 77%), C-: [70%, 73%), D: [60%, 70%), F: [0%, 60%). These percentages are guidelines for both the student and instructor and may be changed as needed to reflect circumstances in the course. Any changes that occur are likely to be minor.

Academic Honesty

Plagiarism on any lab, assignment, exam, or project will result in 0 points awarded and an immediate referral to the department and university. There will be no exceptions to this rule. Undergraduate Academic Integrity Policy must be followed.

Honest Collaboration

The golden rule of academic dishonesty is that you should not claim to be responsible for work that is not yours. This is obviously open to some interpretation, and you'll be getting some help from me, the internet, other students, and more throughout the course. This is OK and class is encouraged to be welcoming and collaborative. To help (but not entirely define) the bounds of acceptable behavior, there are three important rules for labs/projects:

- **By you alone:** All code that you submit (other than skeleton code) should be written by you alone, except for small snippets that solve tiny subproblems (e.g googling "remove space from string c++").
- **Do not possess or share code:** Before you've submitted your final work for a lab/project, you should never be in possession of any solution code that you did not write. You will be equally culpable if you distribute such code to other students or future students of this course. **DO NOT GIVE ANYONE YOUR CODE – EVEN IF THEY ARE DESPERATELY ASKING. DO NOT POST SOLUTIONS TO LABS/PROJECTS PUBLICLY ONLINE (on a public GitHub repo or anywhere else)!** If you're not sure what you're doing is OK, please ask.
- **Cite your sources:** When you receive significant assistance on a lab/project from someone else, you should cite that assistance somewhere in your source code with the `@source` tag. Use your best judgment as to what constitutes 'significant'.

For clarity, examples of specific activities are listed below:

- **Permitted:**
    - Discussion of approaches for solving a problem.
    - Giving away or receiving significant conceptual ideas towards a problem solution. Such help should be cited as comments in your code. For the sake of others' learning experience, we ask that you try not to give away anything juicy, and instead try to lead people to such solutions.
    - Discussion of specific syntax issues and bugs in your code.
    - Using small snippets of code that you find online for solving tiny problems (e.g. googling "uppercase string c++" may lead you to some sample code that you copy and paste into your solution). Such usages should be cited as comments in your code!
- **Permitted with extreme caution:**
    - Looking at someone else's code to assist with debugging. Typing or dictating code into someone else's computer is a violation of the "By You Alone" rule.

    - Looking at someone else's code to understand a particular idea or part of a lab/project. This is strongly discouraged due to the danger of plagiarism, but not absolutely forbidden. We are very serious about the "By You Alone" rule!
    - Working on a lab/project alongside another person or group of people. Your code should not substantially resemble anyone else's!
- **Absolutely forbidden:**
    - Possessing another student's code in any form before a final deadline, be it electronic or on paper. This includes the situation where you're trying to help someone debug. Distributing such code is equally forbidden.
    - Possessing solution code that you did not write yourself (from online, e.g. GitHub, etc.) before a final deadline. Distributing such code is equally forbidden.
    - Posting solution code to any assignment in a public place (e.g. a public git repository, etc). This applies even after the semester is over.
    - Working in lock-step with other students. Your workflow should not involve a group of people identifying, tackling, and effectively identically solving a sequence of subproblems.

If you find yourself at such a point of total desperation that cheating begins to look attractive, contact me and I can maybe help. Likewise, if this course is causing massive disruption to your personal life, please contact me directly.

Lateness and Quality of Submission

The deadline for each assignment will be included in the assignment itself. In general, assignments are due before the next week's lecture. **Late assignments will not be accepted and will be given 0 points.** However, life happens and unexpected circumstances may arise. If you feel you are unable to complete your assignment on time, please contact me and we can work something out.

- All assignments must be submitted in the **C++** computer programming language; any other language submissions will receive **0** points unless otherwise directly pre-approved by me.
- All assignments must**:**
    - **be submitted in a buildable state**. If you submit code I can't compile, you will immediately receive **0** points; submitting broken code is seen as unprofessional behavior in industry and lazy in academia so please don't do this.
    - **Enable all warning flags (-Wall -Wextra -Wpedantic or /W3 for MSVC) and compile with no warnings**. Any submissions that produce compiler warnings while compiling will be docked 1 point per warning. Again, submitting code full of warnings (without good reason) is seen as unprofessional in industry and lazy in academia.
- Most assignments will be automatically graded using a program I wrote to build, run, and test your submissions. Full points will be awarded for assignments that build, pass the test suite, and show good C++ programming practices. While my script won't run any tests you write for yourself, I *highly encourage* you to write tests of your own.

Miscellaneous

There will be no make-up for any assignments, projects, or exams without prior approval. If you miss an exam or assignment, you will receive **0** points for it. If you have special needs or circumstances that prove difficult to fit around the class, please speak with me. I'm happy to find something that works for you.

Acknowledgments

The course content, including slides and information on this page, is built upon materials made by Josh Hug, Ananth Kalyanaraman, John Foley, Yanjun Li, and Xiaolan Zhang. Those made by Josh Hug are licensed under CC BY-NC-SA 4.0.