CSCI 2720: Data Structures

Course Information

Location	Hardman Hall, Class 101
Meeting Times	Mon, Tue, Wed, and Thu from 1 pm to 2 pm Tue, Wed, and Thu from 2:15 pm to 3:15 pm
Prerequisites	CSCI 1730 (Systems Programming) and CSCI 2610 (Discrete Mathematics for Computer Science) This course relies heavily on both prerequisites.

Instructor Information

Course Instructor	Mohammad Mohebbi
TAs	Neda Abdolhassani and Reshmi De
Instructor Office	Boyd 223A
Office Hours	Mondays from 2 pm to 3:15 pm (after class) Wed. from 3:15 pm to 4 pm (after class)
Email	mohebbi@uga.edu

Course Description

The design, analysis, implementation, and evaluation of the fundamental structures for representing and manipulating data: lists, arrays, trees, tables, heaps, graphs, and their memory management.

The topics listed below are covered:

Abstract Data Structures, Arrays and Array Mapping Functions (AMFs), Linked Lists, Composite Structures, Chronologically ordered lists, Frequency ordered lists, Rings, general tree structure and terminologies, Abstract Binary trees, Expression/Parse trees, Binary Search Trees, Heap data structures, Priority queues (HPIFO), Heap implementation of HPIFO, Quad-trees, Oct-trees, Graphs and networks, Sorting and merging algorithms, Searching algorithms, pattern matching, Efficiency issues (order of complexity), B---Trees, Hashing functions, and a number of illustrative applications.

Learning Outcomes

This course presents a survey of topics in computer data structures most relevant to student studying computer science. At the end of the semester, all students will be able to do the following:

1. Have a solid foundation on Abstract Data Structures.
2. Use recursion as a powerful problem solving technique in design and development of data structures.
3. Analyze the efficiency of data structures.
4. Select the most appropriate data structure for applications and being able to defend the selection.
5. Build data structures and use them as building blocks to form more complex advanced data structures in a hierarchical manner.
6. Implement various types of lists.
7. Develop binary trees; including: abstract binary tree, expression/parse tree,
   AVL trees, …

8. Develop Heap data structures for important applications.
9. Design and implement various methods in creating Priority Queues.
10. Implement Quad-trees and Oct-trees.
11. Develop various searching algorithms.
12. Develop B-Trees of different orders.
13. Design and develop various Hashing functions.
14. Develop Graph-based data structures (adjacency and path matrices, …)
15. Appreciate the importance and significance of data structures in building real world applications.

Required Textbooks and Materials

Data Structures and Their Algorithms
Harry Lewis and Larry Denenberg (L & D)
Addison Wesley
ISBN-10: 067339736X

Grading

Midterm Exam	25%
Final Exam	25%
Class Participation/Quizzes	5%
Class Attendance	5%
Homeworks	15%
Projects	25%

The course final grade will use the following grading scale:

A	90% or above 90%	C+	76% to 77.99%
A-	88% to 89.99%	C	70% to 75.99%
B+	86% to 87.99%	C-	68% to 69.99%
B	80% to 85.99%	D	60% to 67.99%
B-	78% to 79.99%	F	below 60%

The course instructor reserves the right to further curve grades based on overall grade distributions; however, any such curve ***will not*** reduce any student's grade. Students must be officially registered for this course in order to attend class and to receive any grades.

Tentative Exam Dates

Midterm Exam: July 2rd, 1:00 pm to 3:15 pm in our scheduled classroom
Final Exam: August 1nd, 3:30 pm to 6:30 pm in our scheduled classroom

Attendance Policy

You are expected to attend all classes during the semester. If you are not able to attend a class, then it is your responsibility to find out what was covered during the class and to catch up.

If you are absent due to extreme circumstances such as a family emergency, serious illness/injury, or other significant event, please contact your instructor as soon as possible.

Email Policy

To email the instructor or TA, students are required to use their official UGAmail address, and the subject of a student email to an instructor must include a [cs2720] tag. Emails from addresses outside of UGA or emails that are lacking the proper subject tag may not be received or answered. Typically, the instructor and TA will reply to students' emails within 48 hours Monday through Friday. During the weekends and University holidays, emails will be answered at the instructor's or TA's discretion.

Email communication should NOT be seen as an alternative to meeting with the instructor (or the TA) during office hours. Email will not be used to provide private tutorials or to explain material that was covered in classes you missed. If an email question cannot briefly be answered with a reply email, the instructor will indicate to the student that she or he should see the instructor (or TA) during the announced office hours or make an appointment.

Programming Policy

All programming projects and some parts of homeworks must be written in C++, and all C++ source code must compile using the g++ compiler located in /usr/bin on nike.cs.uga.edu (nike). To submit a program, put its source code, its makefile, and its readme file in a single directory (the name of the directory will given in a homework's or project's directions) on nike, and use the submit utility to submit the directory to the cs2720 account on nike. Programs must have a readme file that states what the project does and how to compile and run it. Also, programs must have a makefile that includes at least three working directives: compile, run, and clean. The first and default compile directive should compile all source code, the run directive should run an example of the program, and the clean directive should remove all compiled code.

After submitting a directory, you are responsible for checking if the submission was successful by checking if a file beginning with "rec" was created in the submitted directory. The "rec" file will show a record of all files submitted along with their timestamps. If a "rec" file does not appear in the directory after submitting it, then the submission was unsuccessful, and you must resubmit. If you cannot submit a project or homework via nike, then email your source code to the instructor before its deadline. Students are responsible for saving and backing up their source code. Students are also responsible for checking that their programs compile and run correctly on nike.

Regrading Policy

A regrade request must be made to the instructor or TA during scheduled office hours within one week of the date that the grade was posted in eLC. This rule applies to any grade recorded by the course instructor or TA throughout the semester. Students are responsible for checking their grades on eLC on a regular basis. After the last class lecture, no regrade requests and no late work will be accepted.

Makeup Exam and Final Exam Policies

No makeup exams will be given. If you are absent the day of an exam due to extreme circumstances such as a family emergency, serious illness/injury, or other significant event, please contact your instructor as soon as possible. Under this scenario, you must provide a copy of detailed documentation to your instructor within one week regarding the reason for your absence. The instructor has the right to decide whether or not to excuse an exam absence. Typically, notes from the health center or doctor's office that state something like "the student was seen at this date/time" are not considered as detailed documentation that would excuse an exam absence. If you are excused by the instructor, your grade for the missed exam will be the same grade that you receive on the final exam. If you are not excused by the instructor, your exam grade will be a zero. The final exam must be taken at the scheduled time and place; however, if you must reschedule the final exam per University policy, then you must petition the instructor at least one week before the final exam is scheduled.

Academic Honesty Policy

All students are responsible for maintaining the highest standards of honesty and integrity in every phase of their academic careers. The penalties for academic dishonesty are severe and ignorance is not an acceptable defense. *You must abide by the University Honor Code and the Academic Honesty Policy. All academic work must meet the standards contained in "A Culture of Honesty". Students are responsible for informing themselves about those standards before performing any academic work.*

In addition, you are expected to have read and understood the CS Academic Integrity policies. Furthermore, all of your course work, including homeworks and projects, are to be strictly your own work. You can talk over general principles and concepts about an assignment with others, but copying, sharing materials, or even looking at, another student's solutions or code, either on paper, on the computer or whiteboard, is not allowed.

CS Academic Integrity policies:

- Read "A Culture of Honesty", the UGA academic honesty policies and CS Academic Integrity policies.
- You must not allow others to copy or look at your work.
- Do not write code for others, and don't have others write code for you. All the programming assignments in CSCI 1301 are to be your own work unless otherwise explicitly stated on the assignment.
- You must not give/share your lab/project assignment work to a fellow student.
- Copying significant portions of code from a fellow student or any other source (including internet) is plagiarism and will be dealt with as such.
- If you have questions about an assignment or if you run into problems, contact your instructor/lab instructor.
- During exams, no assistance and no additional materials are allowed.

All of your coursework must meet the aforementioned policies and rules. Students that violate any of these rules or the UGA Academic Honesty policies will be liable to a penalty. The instructor will strictly enforce Academic Honesty policies and report any violation of the aforementioned policies and rules. Furthermore, the instructor and TA have the right to run plagiarism detection software on submitted work throughout the semester.