CS 171: Introduction to Computer Science II

Fall 2019

Instructor: Ting Li

Email: tli41@emory.edu

Office: Pierce Hall 127

Hours: MWF 1:00pm – 1:50pm

Location: Pierce Hall 104

Office Hour: F 2:00pm – 4:00pm, or by appointment. Drops-in are encouraged if the door is open. You can always knock the door but I may ask you come back at another time. The optimal solution is always to come during the office hours or email me to schedule an appointment.

Textbook: There are no required textbook for this class. If students wish to have a text for reference, Daniel Liang's *Introduction to Java Programming and Data Structure*, and Robert Sedgewick Kevin Wayne’s *Algorithms* are the great resources – but the texts will not be explicitly addressed in the course. Another resources is an online interactive textbook introduction to Java Programming from Zybooks. You can sign up and subscribe it with code “EMORYCS170liFall2019”.

Other required materials: You will, of course, need access to a computer in class. Having a flash drive to regularly backup your work is also highly recommended.

Overview:

This course is a continuation of CS170. Emphasis is on the use and implementation of data structures, and fundamental algorithms, with introductory algorithm analysis, and object oriented design and programming with Java.

Students will be given many opportunities to write programs to demonstrate their mastery of the algorithms and data structures covered in this course. Problem solving and real- world applications will play an important role, providing a driving motivation for developing and/or selecting appropriate algorithms or data structures to accomplish the associated goals as efficiently as possible.

Prerequisites: Successful completion of CS 170 or an equivalent course.
Course objectives: Students at the conclusion of this course should be able to...

- Implement and use various abstract data types, including (but not limited to) stacks, queues, linked lists, binary search trees, heaps, hash tables, sets, and graphs to accomplish various tasks
- Analyze and measure the algorithms using Big O, Ω, and Θ notation, in both the average and worst-case scenarios
- Implement a variety of sorting algorithms, including selection sort, insertion sort, merge sort, quicksort, and heap sort, understanding the advantages and disadvantages of each
- Implement and use data structures and algorithms related to graph theory to accomplish various tasks

Course expectations:

Classes: Class time will be used for short lectures, design examples, in class assignments, and exams. There is no requirement for attendance, it is essential that you come to the class on time every day. Students are responsible for all material covered in class. Missed work, quizzes, or exams will receive a grade of zero.

Exams: You will have three exams and a final exam this semester. The tests will emphasize reading, understanding, and debugging code -- more than writing code. That said, some questions on the tests might also require students to write code. Doing well on these exams will strongly correlate to having read and understood the readings online and other reference material provided, and having worked in earnest -- and successfully -- on the programs assigned up to that point in the class. All the exams including final are closed book and notes.

In general, makeups are not allowed for exams and assignments. However, you have a valid reason for a makeup exam, inform me as soon as possible. Valid reasons include medical emergency, a death in the family, or religious observations. Extensions will only be granted for emergency situations.

Grading: Your grade for this course will be calculated as follows.

| Assignment | 35% |
| Test       | 3*15% |
Final Exam 20%
Total 100%

**Special Accommodations:**

Access, Disability Services and Resources (ADSR) works with students who have disabilities to provide reasonable accommodations. In order to receive consideration for reasonable accommodations, students must contact ADSR and complete the registration process. Faculty may not provide disability accommodations until an accommodation letter has been processed; accommodations are not retroactive. Students registered with ADSR who receive a letter outlining specific academic accommodations are strongly encouraged to coordinate a meeting time with their professor to discuss a protocol to implement the accommodations as needed throughout the semester. This meeting should occur as early in the semester as possible. Contact Access, Disability Services and Resources for more information at (770) 784-4690 or adsroxford@emory.edu. Additional information is available at the ADSR website at http://equityandinclusion.emory.edu/access/students/index.html.

**Honor code policy:**

All work done in this class is governed by the Oxford College Honor Code.

Students may not give, access, or receive any information not expressly permitted by the instructor on tests or exams. Collaboration between students on tests and exams is strictly prohibited.

For the labs and final project, however, students may work in pairs if they wish, unless otherwise directed.

**Religious Holidays:** Instructors are encouraged, not required, to accommodate students’ academic needs related to religious holidays. Please make every effort to negotiate your religious holiday needs within the first two weeks of the semester; waiting longer may compromise your instructor’s ability to extend satisfactory arrangements. If you need guidance negotiating your needs related to a religious holiday, the College Chaplain, Rev. Lyn Pace, ppace@emory.edu, Candler Hall
202, is willing and available to help.

**Please be aware that Rev. Pace is not tasked with excusing students from classes or writing excuses for students to take to their professors.**

Emory’s official list of religious holidays may be found at: http://www.religiouslife.emory.edu/faith traditions/holidays.html.

*This syllabus is a guide for effective learning in this class; it is not a legal contract. The instructor reserves the right to modify the syllabus as needed.*