

CMSC 3890

The Coding Interview

FALL 2020

Course Description

This course provides a comprehensive, practical introduction to technical interviews. The course will start with basic topics such as Big O and String Manipulation. We will then move into more complex topics such as Graphs and Dynamic Programming. Most of the classes will be in-class interviews to give real interview practice.

Course Details

Prerequisites: Minimum grade of C- in CMSC216 and CMSC250

Recommended co-requisite: CMSC351

Credits: 1

Seats per Section: 30

Language(s): Python

No prior knowledge of Python is expected. Students might have the option of using other languages in in-class interviews or assessments, but Python is strongly recommended for technical interviews. All submit server homeworks will use Python.

Textbook (recommended):

[Cracking the Coding Interview](#) by Gayle Laakmann McDowell

Supplementary Material:

All topics [Elements of Programming Interviews](#) by Adnan Aziz, Tsung-Hsien Lee, Amit Prakash
Big O [Competitive Programmer's Handbook sec. 2.2, 2.3](#) (online book) by Antti Laaksonen

Time and Location:

Section	Time	Location
TBD	TBD	TBD
...

Contact:

We will interact with students outside of class in primarily via Piazza. For example, if you are missing class because of an interview or wish to set up a mock interview with any of the TAs, make a Piazza post visible to instructors only. If you have a general question that other students might be able to answer, make a Piazza post visible to everyone. The exception to this is exam scheduling, which will most likely be done using Google Sheets.

Course Facilitator(s):

Head TA	Dhruv Mehta	dhruvnm@umd.edu
	TBD	Naveen Raman nav.j.raman@gmail.com
	TBD	TBD

Faculty Advisor: Tom Goldstein tomg@umd.edu

Schedule (tentative)

Wk	Date	Topic	Point People
1	2020-09-04	Intro / Resumes / Mock Interview	TBD
2	2020-09-11	Arrays & Strings	TBD
3	2020-09-18	Sorting & Searching	TBD
4	2020-09-25	Linked Lists	TBD
5	2020-10-02	Inheritance	TBD
6	2020-10-09	Stacks, Queues, & Heaps	TBD
7	2020-10-16	CMSC389O Midterms Week	
8	2020-10-23	Graphs	TBD
9	2020-10-30	Trees & Tries	TBD
10	2020-11-06	Dynamic Programming	TBD
11	2020-11-13	Technical Concepts	TBD
12	2020-11-27	No class, Thanksgiving Recess	
13	2020-12-04	CMSC389O Finals Week I	
14	2020-12-11	CMSC389O Finals Week II	
16	2020-12-18	Final Exams Week	

Table 1: Schedule for the semester, broken down by week

Please note if you have any questions about a particular weeks pre-lecture activity, lecture activity, homework, or extra credit, the point people for that topic are the best people to answer your question.

Grading

Grades will be maintained on ELMS. You will be responsible for all material discussed in lecture as well as other standard means of communication (Piazza, ELMS announcements, email, etc). This includes deadlines, policies, and assignment changes.

Any request for reconsideration of any grading on coursework must be submitted within one week of when it is returned. No requests will be considered afterward.

Your final course grade will be determined according to the components detailed below. **In addition, ten (10) extra credit opportunities will be provided throughout the semester. In aggregate, the extra credit assignments can boost your grade up to 5%.**

Breakdown

Class Participation (30%)

Most Classes will consist of in-class partner interviews. **Showing up more than 5 minutes late will result in a grade of 0 for participation for that class period.** Students with excused absences will not be penalized for missing class. Please see below for absences policy. **Students with special circumstances, such as a far-away previous class, should speak with instructors on the first day.**

Pre-Lecture Videos/Quizzes (10%)

Students will be responsible for watching pre-lecture videos and completing pre-lecture quizzes or activities to demonstrate their understanding of the content in the videos.

Homework (20%)

Weekly homework assignments will consist of solving coding interview questions and submitting solutions to the UMD CS submit server. Students will be graded on passing test cases, on the time and space complexities of their solution, and on completion of a short write-up regarding their solution. *Note that these are not mutually exclusive: solutions with higher time/space complexities may fail to pass some tests on the submit server — this is intentional.*

Homeworks will be accepted up to 24 hours after the deadline, with a 20% deduction in credit. No homework will be accepted more than 24 hours late.

Midterm — Interview (20%)

The midterm will be a 30-minute Google Hangout technical interview with one of the student facilitators. Students will be expected to solve 1–2 coding questions and have a brief conversation about their experiences and skills.

Final — Interview (20%)

The final will be a 45-minute Google Hangout or in-person (limited availability) technical interview with one of the student facilitators. Students will be expected to solve 1–2 coding questions, have a brief conversation about their experiences and skills, and discuss some technical concepts.

Excused Absence and Academic Accommodations

See the section titled “Attendance, Absences, or Missed Assignments” available at [Course Related Policies](#). **Note that absences due to internship/job interviews will be excused.**

Disability Support Accommodations

See the section titled “Accessibility” available at [Course Related Policies](#).

Academic Integrity

Note that academic dishonesty includes not only cheating, fabrication, and plagiarism but also includes helping other students commit acts of academic dishonesty by allowing them to obtain copies of your work. In short, all submitted work must be your own. Cases of academic dishonesty will be pursued to the fullest extent possible as stipulated by the [Office of Student Conduct](#). It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit <https://www.shc.umd.edu/>.

Also note that any “hard coding” in a homework assignment may result in a score of zero for that assignment. Hard coding refers to attempting to make a program appear as if it works correctly, when in fact it does not. One example of hard coding would be printing the desired output instead of computing it. This is only one example, and if you have any questions as to what constitutes hard coding, be sure to ask ahead of time.

Course Evaluations

If you have a suggestion for improving this class, do not hesitate to tell the instructor or TAs during the semester. At the end of the semester, please do not forget to provide your feedback using the campus-wide CouseEvalUM system. Your commends will help make this class better.

Notes

Syllabus subject to change.