Computer Science
Acing the CS Technical Interview
Friday, 2:00 - 2:50 p.m., CSIC 1122

Sujith Vishwajith, Phong Dinh, Damien Nagle, Jake Bacher

cmscinterviews@gmail.com
www.cmscinterviews.github.io

Office Location: TBD
Office Hours: TBD

This syllabus is tentative and may change.

Course Description: This course presents a comprehensive introduction to the techniques of for solving computer science technical interview problems. General problem types include but are not limited to: arrays, strings, dynamic programming, recursion, trees/graphs, hash-maps, etc. The goal of the course is to help students ace technical interviews and get a job at their dream company.

Prerequisite: CMSC 132.
Note(s): This course won't explain data structures & algorithms covered in CMSC 132 in depth but rather how to implement and use them to solve interview problems.
Credit Hours: No Credits

Instructors:

- **Sujith Vishwajith:** svishwaj@terpmail.umd.edu
- **Phong Dinh:** pdinh@umd.edu
- **Damien Nagle:** dnagle@terpmail.umd.edu
- **Jake Bacher:** jakebacher@yahoo.com

(Optional) Text: *Cracking the Coding Interview*, 6st Edition
Author(s): Gayle Laakmann McDowell; **ISBN-13:** 978-0984782857

Course Objectives:
At the completion of this course, students will be able to:

1. Understand and solve a majority of interview problems
2. Quickly identify approaches to solving an interview problem
3. Explain the thought process while coding solutions
4. Implement approaches through the use of data structures and algorithms
5. Compare efficiencies of different algorithmss
6. Explain code thoroughly and concisely to interviewers
7. Scale code to tackle large space problems efficiently

Course Policies:

- **General**
  - Students are encouraged to take notes during class.
  - Quizzes (Not graded) and mock interviews are closed book, closed notes.
  - Laptops are allowed but not required.
- **Grades**
  - There will be no assigned grades for this course as it is uncredited but students are encouraged to complete assignments on their own time to learn the material.
- **Assignments**
  - Students are expected to work independently first or with small groups to understand the material. **Offering** and **accepting** solutions from others will severely impact the understanding of the material so it is not encouraged. Discussion amongst students is encouraged.
  - **All assignments solutions will be posted on the course website**.
- **Quizzes**
  - Quizzes are optional and self graded.
  - **Short one interview question based pop quizzes will be distributed at the beginning or end of lecture to check understanding of material**.
- **Attendance and Absences**
  - Attendance is expected but optional.
  - Students can find missed handouts on the course website and can schedule appointments with instructors to answer any questions on missed material.

Piazza:

We plan to use Piazza (www.piazza.com), a question-and-answer system designed to streamline discussion outside of the classroom. It supports LaTeX, code formatting, embedding, of images, and attaching of files. It will be moderated by the instructors, but students are encouraged to answer questions.

Extra Material:

Websites such as LeetCode (www.leeetcode.com), GeeksforGeeks (www.geeksforgeeks.com), and TopCoder (www.topcoder.com) are excellent sources for additional practice aside from the worksheets given in class. Many medium/hard level problems are included in Cracking the Coding Interview along with thoroough explanations and solutions.

Schedule on next page.

Week	Content
Week 1	• Problem solving approaches & Resume Design • Reading assignment: Introduction and Chapter 1: Arrays & Strings • Assigned Problem Set
Week 2	• Arrays & Strings / Big-O • Reading assignment: Chapter 2: Linked Lists • Assigned Problem Set
Week 3	• Linked Lists • Reading assignment: Chapter 3: Stacks & Queues • Assigned Problem Set
Week 4	• Stacks & Queues • Reading assignment: Chapter 4: Trees & Graphs • Assigned Problem Set
Week 5	• Trees & Graphs • Reading assignment: None • Assigned Problem Set
Week 6	• Mock Interview • Reading assignment: Chapter 8: Recursion & Dynamic Programming • No Assigned Problem Set
Week 7	• Recursion & Dynamic Programming Pt. 1 • Reading assignment: Chapter 8: Recursion & Dynamic Programming • Assigned Problem Set
Week 8	• Recursion & Dynamic Programming Pt. 1 • Reading assignment: Chapter 10: Sorting & Searching • Assigned Problem Set
Week 9	• Sorting & Searching • Reading assignment: Chapter 5: Bit Manipulation • Assigned Problem Set
Week 10	• Bit Manipulation • Reading assignment: Advanced Topics • Assigned Problem Set
Week 11	• Advanced Topics • Reading assignment: None • Assigned Problem Set
Week 12	• Mock Interview • Reading assignment: None • No Assigned Problem Set