

# 11-601 Coding & Algorithms Bootcamp

## Fall 2024 Syllabus (2024-08-23)

### Course Description

The goal of this course is to enhance your *existing* programming skills by ingraining as deep a mastery of fundamental algorithm and coding skills as possible in the timeframe of the course. We will seek specifically to improve your algorithmic problem solving skills, improve your ability to plan and describe solutions to problems, develop further your ability to translate your thoughts into code and explain that code to others, and thus maximize your chances of superior performance in any coding interview.

### Prerequisites

Students should **already be able to program in Java** (for example, have taken an introductory Java course). While useful, no prior experience with JavaScript or Python is necessary. A laptop is required for the in-lecture coding exercises and exams using Codio.com (see page 10).

### Schedule

Class meets Tuesdays and Thursdays from 11:00am to 12:20pm ET in WeH 7500.

### Instructor

Ralf Brown ([ralf@cs.cmu.edu](mailto:ralf@cs.cmu.edu), GHC 5713) For office hours, see the course Canvas calendar or [www.cs.cmu.edu/~ralf](http://www.cs.cmu.edu/~ralf), or send email for an appointment.

### Teaching Assistants

Ankush Babbar <[ababbar@andrew.cmu.edu](mailto:ababbar@andrew.cmu.edu)>, Paramjit Baweja <paramjib>, Ajay Mittur <amittur>, Shaurya Singh <shauryas>, Riya Singhal <riyapras>, Rajeev Veeraraghavan <rveerara>, Chentianye (Glenn) Xu <[chentiax@andrew.cmu.edu](mailto:chentiax@andrew.cmu.edu)>

### Required Materials

“Cracking the Coding Interview: 189 Programming Questions and Solutions” 6<sup>th</sup> Edition (green cover, ISBN 978-0984782857) by Gayle Laakmann McDowell (~\$39). Available from your favorite bookseller, and at [www.amazon.com/dp/0984782850](http://www.amazon.com/dp/0984782850) . The Indian edition is acceptable if you already have it, but see the FAQ on Piazza for caveats.

Paid account at codio.com (~\$48) after the second lecture of the course.

Other material will be provided as needed during the course (see the Canvas “Modules” page; the relevant modules are listed in the course schedule on pages 7 and 8).

### Waitlist / Auditing

Due to the extreme demand for this course, it is unlikely that you will be enrolled if you are not already on the waitlist by the first day of the semester (i.e. Monday, August 26<sup>th</sup>). Requests to audit the course will not be granted, as enrollment has already been expanded as much as it is possible to handle.

## Workload

Expect to average a full 12 hours per week on the work for this course

- weekly mock technical interviews: 2 hours
- weekly homework assignments: 4-8 hours (see the graph on Page 9)
- plus readings and working through textbook questions

## Other

- There is no substitute for hands-on learning with algorithms and coding. For this reason, the majority of material you will be expected to learn will be covered in depth with the readings and programming assignments you will complete outside lecture. Please complete all readings prior to the corresponding lecture – they are critical to doing well on the per-lecture exercises.
- Codio is integrated with Canvas, so exercises and homework assignments will appear on your Canvas To-Do list.

## Learning Outcomes

Students successfully completing the course should be able to:

- program in Java, Python, and JavaScript
- test and optimize code
- analyze the computational complexity of algorithms
- know how and when to use data structures such as strings, arrays, lists, stacks, FIFO queues, priority queues, hash maps, trees (binary, k-ary, B-Trees, red-black, AVL, splay, etc.), and graphs
- know how and when to use algorithms such as searches (binary, breadth-first, depth-first, Dijkstra's, heuristic, A\*, iterative deepening), sorts (Quicksort, Merge sort, Insertion sort, Selection sort, Heap sort, Radix sort, Counting sort, Tree sort, etc.), recursion, and dynamic programming
- describe the characteristics of various programming paradigms such as object-oriented, functional, and logic programming
- describe design patterns such as Singleton, Factory, Strategy, Adapter, etc.
- work with parallel processing and concurrent data structures
- describe their analysis of a problem and its solution to others

## Evaluation

Grading will be based on the following criteria:

### **Exercises: 20%**

There will be a coding exercise usually lasting 8-12 minutes at the end of each lecture, using Codio.com and accessed via Canvas assignments. Exercises can not be made up if missed, but the four lowest scores will be dropped. *In the event that the course converts to remote-only, lectures will be recorded and the timing of the exercises will be adjusted to accommodate students in other timezones.*

### **Homework Assignments: 30%**

There will be an assignment using Codio.com every week except for the weeks in which there are exams, usually due at the end of the day on the following Monday. Extensions to the due date will only be given in case of a major emergency, and **must** be requested in writing before the assignment is due.

### **Mock Technical Interviews: 25% (Peer 15%, Instructor/TA 10%)**

Each week, every student will be required to administer a peer technical interview lasting 45-55 minutes, as well as be interviewed by another student. Interview assignments will be made by “the shuffle” which will randomly assign specific interviewers/interviewees as well as specific questions. The shuffle will change each week. Students are responsible for arranging and coordinating these interviews once the interviewer/interviewee pairings are announced. If an interview cannot be completed for any reason, a written explanation of the missed interview must be submitted. Forms for the administration and assessment of interviews will be provided and are to be uploaded to Canvas following the interview. The questions for each shuffle (drawn from the textbook) will be on the current or recent subject of study. 30-minute technical interviews with a TA or the instructor are assigned **in addition to** the weekly peer interview beginning in Week 7; there will be two such interviews over the course of the semester. Peer shuffle #1 will be online; peer shuffles #2 and #3 will be face-to-face; and the remainder will be participants' choice.

The 15% for peer interviews breaks down as follows: 5% for submitting interviewee evaluations, 5% for submitting interviewer evaluations, 3% for scores obtained on interviewee evaluations (the highest and lowest score will be dropped), and 2% for scores obtained on interviewer evaluations (the highest and lowest score will be dropped).

### **Exams: 25% (9%, 9%, 7%)**

Three written exams that will take place during class period in Weeks 5 and 11 and during Final Exams using Codio.com and cover all material discussed since the previous exam. Includes multiple choice, short answer, and code questions. Coding will be in Java, Python, and JavaScript, respectively, for the three exams. Students will be required to sign an attendance sheet or submit a signed honor pledge (if the course becomes remote-only) before each exam.

## Evaluation (continued)

### Curving

As this class is **deliberately designed to be challenging**, the target for the class average score is 85% (it was ~83% in 2016 and ~87% in 2017). We will rescale the numerical scores to shift the actual average to 85% (if the unscaled average is below) or 88% (if above) with a standard deviation of 5%. See the grade distribution graph below. Canvas will be updated regularly with the current curve starting with Homework #3.

### Pass/Fail

For those selecting the Pass/Fail option as permitted by their academic program, the criterion for a Pass is a grade of at least 75% *after* curving (a C, the standard minimum passing letter grade for a Masters-level course, would be at least 74% after curving). This is typically achieved with an uncurved score in the high 60s.

### Late Policy

Due dates for assignments will be provided at the time the assignment is given. Late homework and shuffle-interview assignments will be penalized 10% for every day they are late; **assignments will not be accepted more than five days late**, nor after the end of the day on December 9<sup>th</sup> (the Monday after the last day of classes). Late per-lecture exercises **will not** be accepted. Most homework assignments will be due at the end of the day on the following Monday, while shuffle interviews will usually be due at the end of Wednesday the week after they are assigned.

### Regrade Requests

If you feel that the grade given on an assignment should be reconsidered, please submit the regrade request in writing by email to the instructor or in a private posting on Piazza, with a brief description of why you would like the assignment reviewed (e.g. the posted score from a Codio.com assignment does not match your recollection of the number of test cases passed, which can happen due to timing jitter when a test case is close to exceeding the time limit).

## Academic Integrity Policies

Collaboration in learning concepts is encouraged among students – feel free to share notes (**but not code**) and hold study groups. **YOU MUST INDICATE ON EACH ASSIGNMENT EVERYONE WITH WHOM YOU COLLABORATED** (study group, asking for or providing assistance in understanding the underlying material, etc.)

### **\*\*\* Prohibition on sharing or copying code \*\*\***

You **may not share your code** with other students, **use code from another student**, or **use code found online, including code generated by an AI system** such as GitHub CoPilot, Google Bard, or ChatGPT. Some of the assignments used in this course have been used in previous editions of the course or in other courses, and solutions may be available on the Internet. It is explicitly forbidden to search for these problems or their solutions on the net. The plagiarism detector *will* find copying from code submitted in previous years as well as near-duplicate submissions in the current class.

### **\*\*\* Requirement for attribution \*\*\***

You must attribute any code not created and written **entirely** by yourself (e.g. based on collaboratively-produced pseudo-code); this is most easily done with a comment preceding that code which states something like “based on an algorithm description found at <http://foo.com/bar/>” or “pseudo-code developed with <studentname>”.

### **\*\*\* Duty to safeguard your work \*\*\***

*If a student obtains access to another student’s work (even without their knowledge), both students will still be held accountable and punished. **Secure your work!*** To protect future students (including your future self), **do not post your solutions publicly**, even after the end of the course.

### **\*\*\* Penalties \*\*\***

Failure to list all participants of a study group on assignments, to disclose all the participants on a collaborative work, or to disclose the source of any code not written by yourself is not acceptable. These and any other Academic Integrity violations will result in a **minimum** penalty of a **reduction of your final grade** by one full letter grade (e.g. A- to B-) on the first offense, and in the assignment of a failing grade for the course on the second offense. Violations will be reported to your department, which may **impose additional penalties** up to dismissal from the university.

**Violations of academic integrity are very serious and can result in heavy penalties up to suspension or expulsion from the university. Make sure you review and understand the information at:**  
<http://www.cmu.edu/policies/student-and-student-life/academic-integrity.html>.

## **Statement of Support for Students' Health and Well-Being**

Take care of yourself. This course is a **lot** of work. Do your best to maintain a healthy lifestyle this semester by eating well, exercising, avoiding drugs/alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress, especially in such uncertain times as this year.

If you find yourself struggling with the material or workload, please **ask for help**. There are many helpful resources available on campus – including your course instructor and TAs – and an important part of the college experience is learning how to ask for help. It is usually better to ask for support sooner rather than later.

CMU's [Student Academic Success Center](#) offers coaching on many topics which may be helpful for this course, including time management, avoiding procrastination, techniques for understanding course content, and applying for jobs and internships.

**Don't panic if you mess up on one assignment** (we actually expect everyone to be unable to finish some of the lecture exercises). Your grade in this course is based on a very large number of scores, so any individual assignment has very little effect on your final grade.

*If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS) is here to help: call [412-268-2922](tel:412-268-2922) and visit <http://www.cmu.edu/counseling/> You will not be alone – one in eight CMU students make use of CaPS each year.*

*If you or someone you know is feeling suicidal or in danger of self-harm, call someone immediately, day or night:*

CaPS: [412-268-2922](tel:412-268-2922)

Re:solve Crisis Network: [888-796-8226](tel:888-796-8226)

*If the situation is life threatening, call the police:*

On campus: CMU Police: [412-268-2323](tel:412-268-2323),      Off campus: 911

If you are worried about affording food or feeling insecure about food, there are resources on campus who can help. Email the CMU Food Pantry Coordinator to schedule an appointment:

Pantry Coordinator [cmu-pantry@andrew.cmu.edu](mailto:cmu-pantry@andrew.cmu.edu)

412-268-8704 (SLICE office)

# Course Schedule

Wk	Date	Topic	Readings	Textbk Questions
1	8/27	Course Overview Java - Strings and Arrays	McDowell Ch. 1 Canvas: "Strings/Arrays"	1.1-1.9
	8/29	Computational Complexity String Manip., Regular Expressions	McDowell Ch. VI and 2 Canvas: "Complexity"	2.1-2.8
2	9/3	Interview Techniques Java – Testing Defensive Programming	McDowell Ch. VII, 11 Canvas: "Testing" You Are Not Done Testing... (p. 2, 31-32)	11.1-11.6
	9/5	Searching: Binary Search, BFS/DFS Attacking a Coding Problem	McDowell Ch. 3 and Ch. 10 Canvas: "Searching"	3.1-3.6
3	9/10	Searching 2: Path Searches	McDowell Ch. 10	10.3-10.5
	9/12	<b>No Class*</b> – STEM Career Fair	McDowell Ch. 4 Canvas: "Recursion"	4.1-4.12
4	9/17	Recursion Java Code Optimization	McDowell Ch 8 and Ch. XI, p. 633-636	8.3, 8.4, 8.6-8.9
	9/19	Dynamic Programming Hash Tables	Canvas: "Dynamic Programming"	8.1, 8.2, 8.11, 8.13
5	9/24	Java – Bit Manipulation	McDowell Ch. 5	5.1-5.8
	9/26	<b>Exam 1</b> (Java)		
6	10/1	Python Syntax	Canvas: "Python" Whirlwind Tour...Python	16.4-16.6, 16.11, 16.17
	10/3	Python – String Parsing and Manipul. Python – Regular Expressions Graphs, Cycle Detection		
7	10/8	Sorting: QuickSort, Merge Sort No Class? * – Grace Hopper Conf	Canvas: "Sorting"	10.1, 10.2, 10.6, 10.11
	10/10	Sorting 2: Insertion/Radix/Bucket, ...		
	10/15	<b>No Class</b> – Fall Break		
	10/17	<b>No Class</b> – Fall Break		
8	10/22	Object-Oriented Prog / Design Patterns	McDowell Ch. 7	7.1-7.12
	10/24	Functional Programming	McDowell Ch XI, p 642-644 Functional Prog in Python	
9	10/29	Python – Testing / Optimization Randomized Algorithms	McDowell Ch 6 Canvas: "Testing"	17.6, 17.10, 17.14

Wk	Date	Topic	Readings	Textbk Questions
		Monkey Patching		
	10/31	Python – Recursion / Dynamic Prog Search: Iterative Deepening, Bidirect'l		
10	11/5	<b>No Class</b> (Democracy Day)		
	11/7	State Machines Combinatorics Priority Queues Game Strategies	Canvas: “State Machine” How I Learned...the State Machine	
11	11/12	Self-Adjusting Trees B-Trees, red-black trees, splay trees...	McDowell Ch. XI, pages 637-642	
	11/14	<b>Exam 2</b> (Python)		
12	11/19	JavaScript - Syntax	Canvas: “Javascript” Intro to JavaScript for Java Progs	
	11/21	JavaScript – Strings and Sorting string searches (KMP, BM, Rabin-Karp)	JavaScript for Java Developers	
13	11/26	JavaScript – Recursion and DP Sorting with Duplicates (Counting, ...) Math and Logic Puzzles		
	11/28	<b>No Class</b> (Thanksgiving)		
14	12/3	JavaScript – Testing / Optimization Parallel Processing and Scalability	McDowell Ch. 9 and 15	
	12/5	Concurrent Data Structures Logic Programming		
	TBD	<b>Exam 3</b> (JavaScript)		

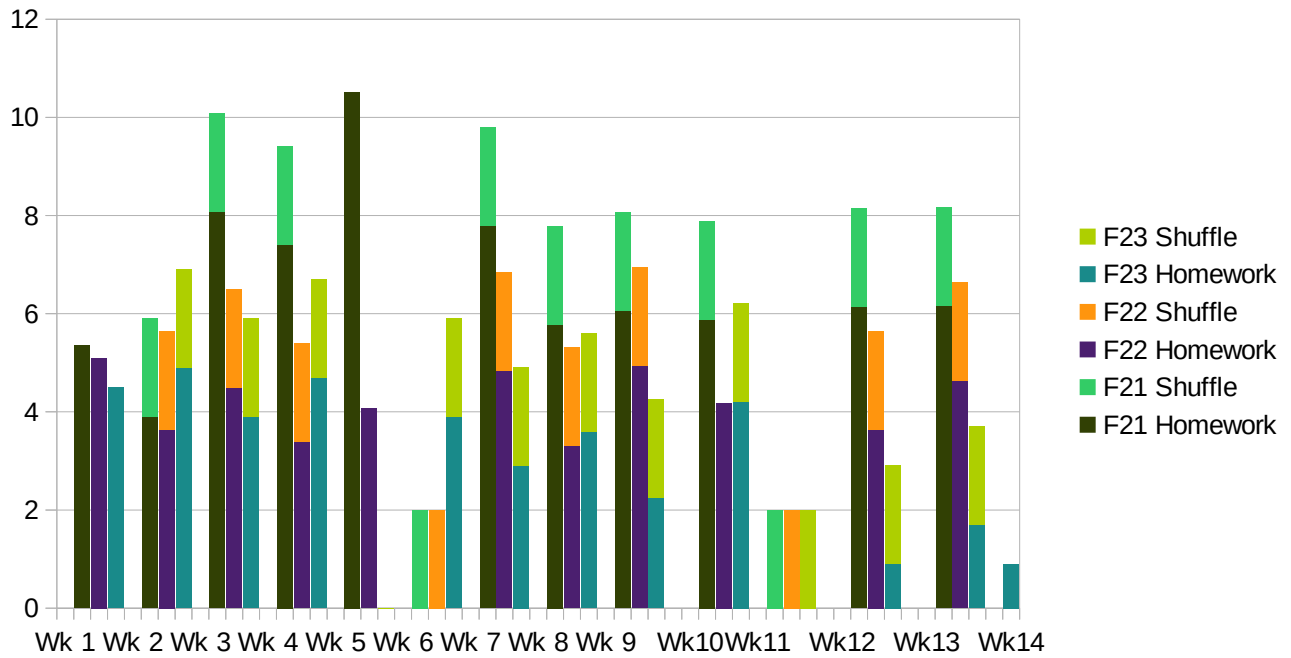
**Homework due dates:** 9/3/2024 (#1), 9/9/2024 (#2), 9/16 (#3), 9/23 (#4), 10/7 (#5), 10/21 (#6), 10/28 (#7), 11/4 (#8), 11/11 (#9), 11/25 (#10), 12/2 (#11), 12/6 (#12).

**Shuffle due dates:** 9/11, 9/18, 10/2, 10/9, 10/23, 10/30, 11/6, 11/20, 11/26, 12/6

**\* Skipped Lecture:** The lecture schedule from 9/12 to 10/10 may vary depending on the exact dates of the STEM Career Fair and a class vote on which of those dates there should not be a lecture so students can attend either it or the Grace Hopper Conference (sorry, we can't skip both days).

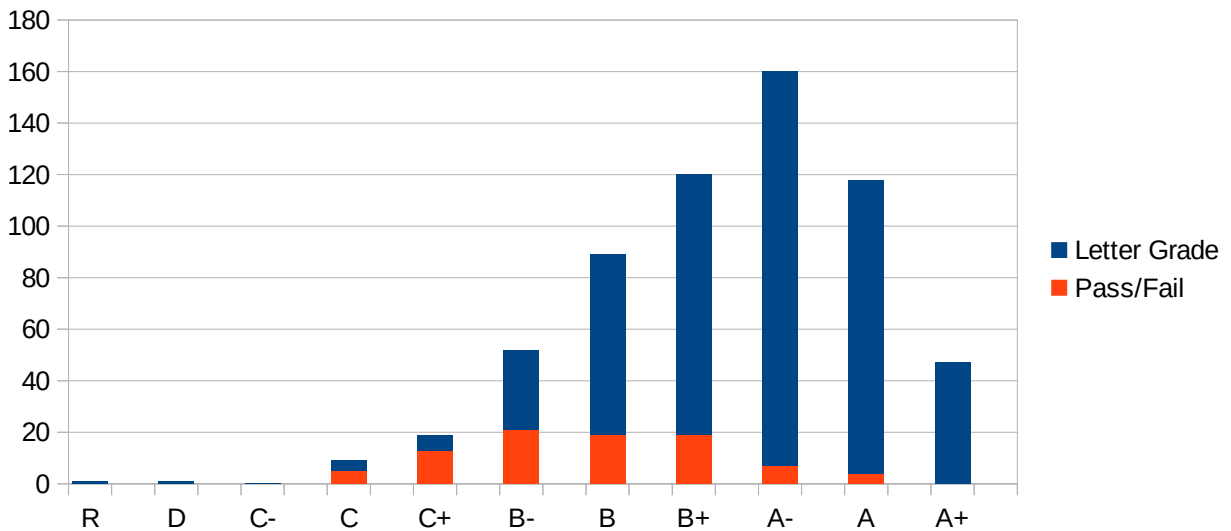


### Median Homework Load in hours, Fall 2021 to Fall 2023 classes



Times above do not include time spent on readings and studying. Schedule adjusted in F23.

### Grade Distribution, Fall 2017 to Fall 2023 classes



Notes: The failure and many of the C/C+s resulted from Academic Integrity Violations. Mid-semester grades were a good predictor of final grades, with the majority of students within one grade step (e.g. A+ to A or B+ to A-). *Past results are not a guarantee of future performance.*

## About Codio

Codio.com will be used for all exercises, homework assignments, and exams. There is a fee of ~\$50 associated with creating an account for use with this course (but you don't need to create the account until after the second lecture).

Codio provides the following features:

- an editor with code completion
- a debugger
- a code visualizer (for some problems – it is limited in the size and complexity of programs supported)
- automatic scoring of submissions based on the number of test cases passed
- timed tests, with automatic submission of code when the time runs out
- code playback – retrieve an older version of your code if your changes made things worse
- plagiarism detection (flagging submissions which are suspiciously similar to each other)
- integration with Canvas

## Frequently Asked Questions

**Q: I was just notified that I'm now enrolled, but I can't access the course in Canvas!**

A: Canvas updates its roster from SIO several times per day rather than instantly. Please be patient – you will normally have access within four hours.

**Q: How do I get my assignment for the peer mock technical interviews (shuffle)?**

A: The information on who to interview, who will interview you, and which questions to ask will be posted to Canvas as a comment on the appropriate assignment in your gradebook.

**Q: What programming language(s) will be used during the interviews?**

A: Instructor and TA interviews will all be in Python (to keep things consistent for everyone). Peer interviews will be in Java for the first half of the semester, then switch to Python and JavaScript. You will be informed which language to use when the interview is assigned in your gradebook.

**Q: My laptop crashed/I fell ill/my Internet went out and the timer ran out on my assignment in Codio. What now?**

A: Send an email and we can give you additional time to make up for the lost time.

**Q: I realized after submitting my assignment that I forgot to cite someone or something. What do I do?**

A: Send an email to your instructor as soon as possible with the missing citation.

**Q: What if I panic under pressure and do something I shouldn't have?**

A: Your penalty will be less severe if you confess before we find your violation ourselves.

**Q: I submitted XYZ, but the Canvas gradebook still shows a missing grade.**

A: Shuffle grade entries are handled by an external script which must be run manually, while Codio grades are posted once manually released after the due date. Shuffle assignments are normally processed shortly after the due date and after midnight on the following few days. If the grade remains missing more than a week after submission, send an email.

**Q: What about ABC...?**

A: See the FAQs posted on [Piazza](#).