11-601 Coding & Algorithms Bootcamp
Tentative Fall 2021 Syllabus (2021-06-04)

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.

Schedule

Class meets Tuesdays and Thursdays from 11:50am to 1:10pm ET in MM A14.

Instructor

Ralf Brown (ralf@cs.cmu.edu, GHC 5711) For office hours, see the course Canvas calendar or www.cs.cmu.edu/~ralf, or send email for an appointment.

Teaching Assistants

TBD

TBD

TBD

TBD

Office hours are posted to the course Canvas calendar.

Required Text

"Cracking the Coding Interview: 189 Programming Questions and Solutions" 6th Edition (green cover) by Gayle Laakmann McDowell. Available through the CMU bookstore, from your favorite bookseller, and at www.amazon.com/dp/0984782850 . The Indian edition is acceptable if you already have it, but see the FAQ on Piazza for caveats.

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 third day of the semester (i.e. Wednesday, September 1st). 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: 4-7 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.
- You will receive email from HackerRank.com when homeworks and exercises are assigned.

Lecture Videos

If a remote-only section is needed, lectures will be recorded. You will be able to access the recordings via the Zoom tab in the course's Canvas site within a few hours after the end of the lecture. Remember that the recordings **may not** be shared **anywhere** to protect everyone's privacy rights under U.S. federal law (FERPA).

Evaluation

Grading will be based on the following criteria:

Exercises: 20%

There will be an exercise usually lasting 8-12 minutes at the end of each lecture. Links to each exercise will be emailed early in the morning prior to the lecture. Exercises can not be made up if missed, but the three lowest scores will be dropped.

Homework Assignments: 30%

There will be an assignment every week except for the weeks in which there are exams, usually due at the end of the day on the following Monday. A homework assignment will usually consist of two HackerRank exercises, each of which will typically allow 12 to 24 hours to complete from the time they are started. 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 normally be two such interviews over the course of the semester.

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%

Three equally-weighted written exams that will take place during class period in Weeks 6 and 11 and during Final Exams using HackerRank and cover all material discussed to date. Includes multiple choice, short answer, and code questions. Coding will be in Java, Python, and JavaScript, respectively, for the three exams.

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 minimum passing letter grade for a Masters-level course, would be at least 74% after curving).

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 seven days late,** nor after the end of the day on December 6th (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 re-grade 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 HackerRank 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**. 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 and alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress.

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. Asking for support sooner rather than later is often helpful.

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 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   _Re:solve Crisis Network: 888-796-8226_

If the situation is life threatening, call the police:

   _On campus: CMU Police: 412-268-2323,   Off campus: 911_

Course Schedule

Wk	Date	Topic	Readings	Textbk Questions
1	8/31	Course Overview Java - Strings and Arrays	McDowell Ch. 1 Canvas:"Strings/Arrays"	1.1-1.9
	9/2	Computational Complexity Java - String Manipulation Regular Expressions	McDowell Ch. VI and 2 Canvas: "Complexity"	2.1-2.8
2	9/7	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
	9/9	Sorting: Merge Sort, QuickSort Attacking a Coding Problem	McDowell Ch. 10 Canvas: "Sorting"	10.1, 10.2, 10.6, 10.11
3	9/14	Searching Binary Search, BFS/DFS	McDowell Ch. 3 Canvas: "Searching"	3.1-3.6, 10.3-10.5
	9/16	Searching (2) Path Searches	McDowell Ch. 4	4.1-4.12
4	9/21	**No Class** - Technical Opp. Conference	McDowell Ch. XI, pages 633-636	10.7-10.10
	9/23	Sorting (2) Insertion/Radix/Bucket, ...	McDowell Ch. 13	13.1-13.3, 13.5
5	9/28	Recursion Java Code Optimization	McDowell Ch. 8 Canvas: "Recursion"	8.3, 8.4, 8.6-8.9
	9/30	Dynamic Programming Hash Tables	Canvas: "Dynamic Programming"	8.1, 8.2, 8.11, 8.13
6	10/5	Java – Bit Manipulation	McDowell Ch. 5	5.1-5.8
	10/7	**Exam 1** (Java)		
7	10/12	Python – Syntax	Canvas: "Python" Whirlwind Tour...Python	16.4-16.6, 16.11, 16.17
	10/14	**No Class** – Mid-semester Break		
8	10/19	Python – String Parsing and Manipul. Python – Regular Expressions Graphs, Cycle Detection		
	10/21	Object-Oriented Prog / Design Patterns	McDowell Ch. 7	7.1-7.12
9	10/26	Functional Programming Python – Sorting	McDowell Ch XI, p 642-644 Functional Prog in Python Canvas: "Sorting"	

Wk	Date	Topic	Readings	Textbk Questions
	10/28	Python – Testing / Optimization	Canvas: "Testing" Monkey Patching	17.6, 17.10, 17.14
10	11/2	Python – Recursion / Dynamic Prog Search: Iterative Deepening, Bidirect'l		
	11/4	State Machines Combinatorics Game Strategies	Canvas: "State Machine" How I Learned...the State Machine	
11	11/9	Self-Adjusting Trees B-Trees, red-black trees, splay trees...	McDowell Ch. XI, pages 637-642	
	11/11	**Exam 2** (Python)		
12	11/16	JavaScript - Syntax	Canvas: "Javascript" Intro to JavaScript for Java Progs	
	11/18	JavaScript – Strings and Sorting	JavaScript for Java Developers	
13	11/23	JavaScript – Recursion and DP Sorting with Duplicates (Counting, ...)		
	11/25	**No Class** (Thanksgiving)		
14	11/30	JavaScript – Testing / Optimization Priority Queues Math and Logic Puzzles	McDowell Ch. 6	
	12/2	Parallel Processing and Scalability Randomized Algorithms	McDowell Ch. 9 and 15	
	TBD	**Exam 3** (JavaScript)		

Homework due dates: 9/6/2020 (#1), 9/13/2020 (#2), 9/20 (#3), 9/27 (#4), 10/4 (#5), 10/18 (#6), 10/25 (#7), 11/1 (#8), 11/8 (#9), 11/22 (#10), 11/30 (#11), 12/3 (#12).
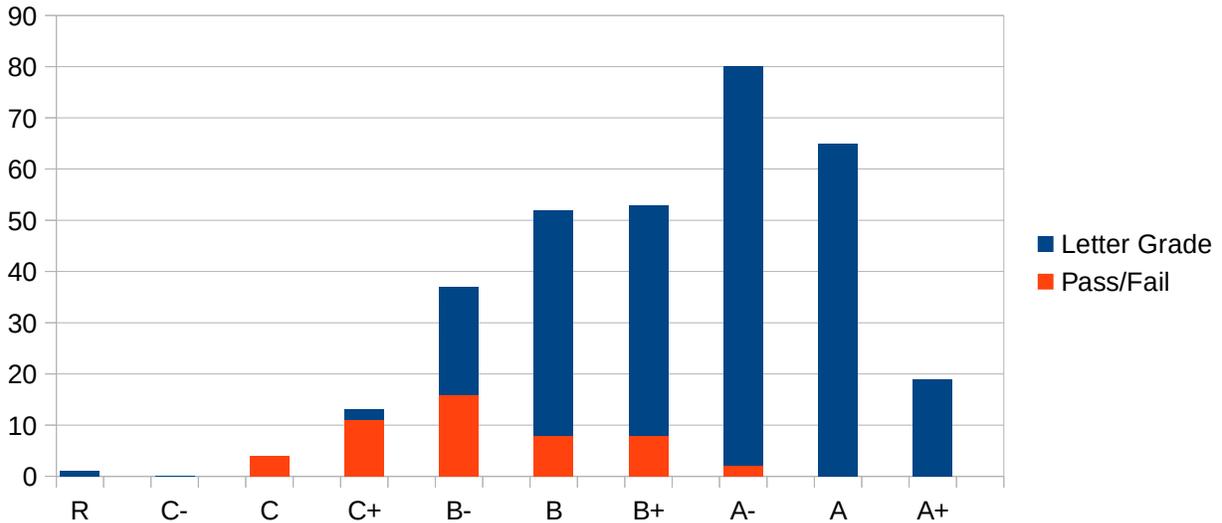**Shuffle due dates:** 9/15, 9/22, 9/29, 10/13, 10/20, 10/27, 11/3, 11/10, 11/17, 11/23, 12/2

Median Homework Load in hours, Fall 2018 to Fall 2020 classes
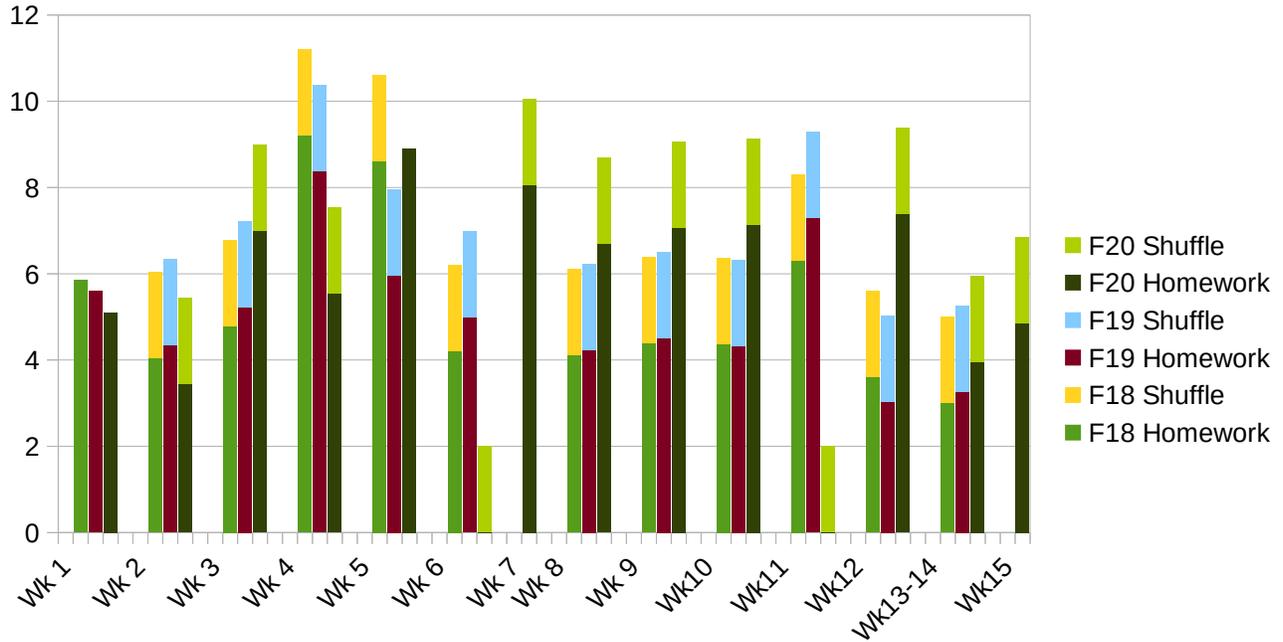
Times above do not include time spent on readings and studying. Schedule changed for F20.

Grade Distribution, Fall 2017 to Fall 2020 classes

Notes: The failure resulted from an Academic Integrity Violation. Mid-semester grades were a very good predictor of final grades, with the majority of students receiving exactly the same final grade as midterm grade and nearly all within one step (e.g. A+ to A or B+ to A-). *Past results are not a guarantee of future performance.*

About HackerRank

Homeworks, exercises, and exams will be conducted using the system provided by
HackerRank.com.

HackerRank provides the following features:

- an editor with code completion

- automatic scoring of submissions based on the number of test cases passed

- timed tests, with automatic submission of code when the time runs out

- logging of activity timelines and all code submissions (if you had a partially-working
  solution but the automatic submission was broken, let us know and we can verify your
  partial solution and give you credit for it)

- plagiarism detection (flagging submissions which are suspiciously similar to each other)

Sample Problem

If you would like to attempt a sample homework problem to see whether this course is for you,
one of the problems from the first homework assignment in the Fall 2016 and Fall 2017 editions
is available at

> http://hr.gs/11601f18sample

This problem took students a median of about 3.5 hours and was generally considered the
second-hardest of the course. You will have 12 hours to complete the problem once you begin.

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'll have access in a few hours.

Q: What is the policy if part of a homework is submitted on time and part late?

A: The late penalty of 10% per day will only be assessed on the points for the part which is submitted late; the part which is submitted on time will receive full points earned.

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 Java (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 homework assignment in HackerRank. 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 transgression ourselves.

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

A: Most of the grade entries are handled by an external script which must be run manually. For example, homework assignments are normally processed shortly after the due date and again shortly after the last possible late submission date. If the grade still shows as missing more than a week after submission, send an email.

Q: What about ABC...?

A: See the FAQs posted on Piazza.