CodingDojo Bootcamp - Syllabus

Dojo Rules:

1. Be Present

The students that consistently struggle and get behind are the ones that show up late and leave early. This is not a 9–5 job where you clock in and out. Algorithms and lectures start each day at 9:00 AM, but we highly encourage you to be here between 8 and 8:20 to give yourself time to get settled in beforehand. Afternoons are unstructured, but those that stay focused for long stretches of time make the most progress. In the evening you may stay as late as you want. If you have no prior programming experience and expect to keep up, we suggest you come in on Saturday to continue your work.

Tardiness / Absence *- Life has a tendency to get in the way - sometimes you might be late to, or even miss, an activity at the dojo. It is your responsibility to get the information that you have missed; we will not chase you. We are able to record the lectures and can do so occasionally if you let us know ahead of time. If you are going to be late, be courteous to the instructor and other students and come in quietly through the back door (if applicable).*

Daily Schedule *(subject to change)*

9:00a	**Algorithms** *(1hr)*
10:00a	**Recap** previous day *(~5-10m) &* New **Lecture** *(~20-30m)*
10:45a	**Group Activity** *(~1hrs) &* Today's **expectations** *(~5m)*
12:00p	Lunch *(1hr)*
1:00p	**Work on Assignments** - Open Lab *(Extra Sessions, Demos, Code Reviews)*
4:45p	**Recap** of the day & **Expectations** for the following day *(~10m)*

2. Strength Through Struggle

This bootcamp is difficult. Make no mistake. You will struggle, frequently at first. It is a part of our learning philosophy that you do struggle. It is in struggling that you will truly learn and become ever increasingly self-sufficient. However, although we want you to find answers on your own, we are here to answer your questions. The goal is to maximize your velocity through our curriculum. Students that ask questions without struggling are not learning, and students who are struggling without asking questions are burning precious minutes, slowing their pace and curbing their understanding. To handle this, we enforce a 20 Minute Rule:

20 Minute Rule - When you begin to struggle:

1. Put yourself first. Utilize your wits, notes, internet resources, etc as your primary resource, but spend ***NO MORE*** than 20 minutes trying to solve a problem on your own.
2. Ask ***at least 2*** of your cohort mates for help. A strong relationship to your cohort mates is another primary benefit of taking an onsite bootcamp. The students around you are the boughs of your new professional network tree.
3. Ask a Teaching Assistant (TA) or instructor. Our TAs are recent graduates of the bootcamp and have been in your very position. Utilize them to help get you unstuck.

The key is to strike the perfect balance between taking it upon yourself to develop strength through struggle, and knowing when to get help in the sake of time.

3. Be Humble

Respect those around you and their learning pace. Do not boast about how far you have gotten or ask questions during lectures that may be far beyond the scope of what is being taught. If someone asks you for help, be courteous and take time to help. In explaining these concepts, you solidify your understanding. Conversely, be respectful of fellow students and don't continuously bother them with questions. Feel free to ask different people in your group or other groups.

Zero Tolerance - We do not tolerate negative attitudes or disrespect in this environment. If we hear or see that you are being a nuisance or disrespectful to others in any way, shape or form, we will kindly ask you to leave the bootcamp.

4. Pair Programming

The power of pair programming The secret to getting the most out of this bootcamp is to work together with your cohort on assignments. Those who pair program are known to complete tasks about twice as fast! You will be less stressed when you work with someone else; also because you must verbalize what you've learned, you will retain knowledge more effectively. Finding a good pair programmer could really help you learn a lot of concepts while eliminating unnecessary stress. Commit to helping others, and commit to working in small groups (2 to 3).

Expectations and Resources:

Daily Schedule *(subject to change)*

9:00a	**Algorithms** *(1hr)*
10:00a	**Recap** previous day *(~5-10m) &* New **Lecture** *(~20-30m)*
10:45a	**Group Activity** *(~1hrs) &* Today's **expectations** *(~5m)*
12:00p	Lunch *(1hr)*
1:00p	**Work on Assignments** - Open Lab *(Extra Sessions, Demos, Code Reviews)*
4:45p	**Recap** of the day & **Expectations** for the following day *(~10m)*

(**First half of the day** - The first half of the day is very structured, and strategically engineered to give you the best amount of resources at the right pace and order)

Algorithms
Morning algorithms are extremely important, always be on-time by arriving before 9:00am. Not only do you learn how computers work (and can be a better developer because of it) but you will also learn algorithms that are required to be demonstrated in interviews for large companies such as Google, Amazon, Microsoft, Facebook and others like them.

Recap & Lecture
The recap will briefly review the previous day's lecture topics and launch into the new lecture. Be sure to have read the chapter and watched the videos before 9:00am. You don't need to fully understand the topics, just read enough to have the context of what is presented so you may interact during the lecture and ask questions.

Group Activity
Always participate in the group activities, they are not optional. They are strategically placed in our program to help you learn concepts in a group setting. This will give you and your peers the chance to discuss the topics and better understand the day's lecture so you are well prepared for the upcoming assignments.

Lunch
Generally we break at 12:00 (noon) for lunch without notice, so you're free to break for lunch right at 12:00 unless there is an event occurring - in that case you will know ahead of time.

(**Second half of the day** - The second half of the day is less structured purposely to allow our instructors to plug in their efforts where they are most needed based on your progress.)

Open Lab (Second half of the day)
Open lab is the time where students get most of their work done through pair programming or self study (we highly encourage you to work in groups, it's much more effective). This is also the time where instructors may inquire for demos and other activities to help you learn concepts and move forward.

Code Reviews
Code reviews are done weekly. We require key assignments to be reviewed with you in person, and offer to code review the rest of the assignments as you deem it beneficial. This is the time where we will look through your code, comment on code quality, and logic. Pay close attention in these sessions, they are extremely helpful. Also, we will request you to have all of these assignments in open tabs on your editor for us to review efficiently.

Recap and Expectations
The instructor will go over the day's topics to ensure your clear as to what you're covering for the day, as well as clearly state the upcoming expectations for the following morning (such as what tabs and videos you should read/watch before the next day).

Belt Exams
Belt exams will be given each month on the subjects covered during that time period. Generally these are given one week before the end of each course, on Friday. If you don't pass the exam, you may re-take it as many times as you wish (before you retake the exam, get help from your peers or instructor to complete your initial exam first). Notify your TA or instructor to schedule a time and date for the retake. During the bootcamp you may not take an exam of a level higher than what you have passed. You will receive a belt sticker for each belt exam passed.
Note: Your goal here at Coding Dojo is to become a self sufficient developer, a way to prove this to us and most importantly to yourself, is successfully achieve the black belt certification (belt exam) on the tracks you learn.

There is usually a question regarding the importance and priority of 'Exams' and 'Projects' during project week. So here is something to consider to better understand our reasoning for always encouraging you to pass your exams before your work on a project:

Completing a group project is great! However, because it is a team effort, you will not build it all yourself, and thus you won't truly know whether you could have done it on your own. Completing a belt exam on your own first, proves to you your self sufficiency and gives you a lot of confidence. Once you have the belt, completing projects will be more rewarding and definitely easier! Exams and projects are both important, but *always* get your belt *before* working on a large team project during project week.

Tools
* *learn.codingdojo.com* - Learning platform with lessons, videos, quizzes, assignments, Q&A forums, answers, etc.
* *algorithm.codingdojo.com* - Online platform for improving your skills and speed in solving algorithm challenges.
* *exams.codingdojo.com* - Take the belt exams here.

Videos
In addition to lectures, you will watch a series of videos for each course. We want to set expectations: our videos cover material quickly; production quality is not always the greatest. Our focus is to ensure you get content you need, as clearly as possible.

Pace
We have created a detailed curriculum with assignments and exams, but due dates are soft and exams are optional. We will do all we can to ensure that you keep up with the curriculum. We highly encourage you to stay on pace with assignments/exams, including as many optional ones as you can, but please don't feel left behind if you are unable to keep up. You will have access to the platform course material for three months following the bootcamp, so you can continue your progress afterward.

Keys *(if applicable)*
Students that are consistently here early and on weekends will be given a key to the classroom. We will track who has each key and re-distribute accordingly. It is your responsibility to get the key back if you lend it out to another student.

Sports Day
Every Thursday, we will go to a nearby park to play sports. We leave at 11:00a and return to the dojo between 12:30 and 1:00. To participate in Sports Day, you must sign a waiver. Sports days consist of games like Volleyball that get the entire class playing together, for team bonding. If you absolutely cannot play sports, we still advise you to take a mental break and spend it taking a walk around the building or just step away from your computer to break your thought process and clear your mind for an hour.

Instructors
Instructors at Coding Dojo have a variety of skills and backgrounds. Some have 15+ years of experience; some are alumni of our program. It's important to note that only the top 1-3% of our graduates get the opportunity to become Apprentice Bootcamp Leaders (a rotational program for individuals who pass a high bar of technical knowledge, cultural fit and ability to present). These alumni are in a training cadence that requires them to build full applications monthly along with the completion of all assignments to perfection before they are deemed ready to co-teach a track with a Sr. Instructor present.
Our Industry hires (those with 15+ years of industry experience) are vetted to ensure that they fit our Coding Dojo culture and embrace our vision of becoming the best place to learn to code. These instructors also go through a rigorous amount of training and observing before we deem them ready to start co-teaching with a Sr. Instructor. All of our instructors, externals and alumni are required to attend the program as students so they may understand the kind of experience our students receive first hand. As you go through the program, you'll have different instructors for each track. This is also by design, we find it is most effective for our instructors to instruct the same track multiple times before swtiching to a different technology stack. All instructors go through a certification process that tests their efficiency, depth of knowledge and ability to teach the content before they are ready to instruct on their own.