Module #3 - Get Your Mind Right (Required)

As every successful person will tell you, "Attitude is everything."

For this reason, as you embark on the challenging endeavor ahead, it's critical that you align your attitudes and expectations with the realities to come. Don't underestimate the amount of mental and emotional discipline this bootcamp will require. Take the time to prepare yourself by reading through the following tips and suggestions.

P.S. Don't breeze through these! They are *important*!

Major Keys to Success

1. A Little Humility Never Hurt Nobody...

Whether you believe it or not, this bootcamp will be one of the most challenging learning experiences of your life. For most of you, the skills being taught in the program will be fundamentally different from any you've been exposed to before. In fact, your past skills, schooling, and experiences are unlikely to help you become a successful developer. Instead, in order to succeed, you will need to channel your inner toddler and *learn to learn* again.

This will be a *very* challenging identity for many of you to take on. If you've had past academic or career successes in another trade or industry, you may find yourself resisting the idea that you can *fail* so dramatically at coding.

Yet, heed this advice now: "A humble learner is the readiest learner." Take each class as a new opportunity to be proven a fool and learn anew. Six months (three if you're in a full-time program) from now, you will marvel at the mastery you gained.

2. There is No Coding Pill...

The greatest self-lie you can tell is that any program will *teach* you to code. Instead, it's important to realize that, in order to succeed, *YOU* must own your learning experience.

Coding, like any craft, requires diligent effort, constant experimentation, and a relentless desire to self-improve. These characteristics cannot be taught in any classroom. From Day 1 through Day 180 and beyond, remind yourself that YOU are responsible for your ultimate success. We will be here to guide, to encourage, to facilitate, to point out your mistakes, and to show you the path, but it's up to you to put in the effort.

Don't get discouraged! We know you can do it.

3. Put in the Hard Hours (At Least 20 Hours!) ...

According to author Malcolm Gladwell, successful people put in a minimum of 10,000 hours of deliberate effort to master their craft. As newcomers to the field of web development, be ready to put in your share of hours.

While the bare minimum to *survive*this program is 10 hours of outside class time, we've consistently found that those most successful put in closer to 20 hours of outside effort per week. At times, this number might even need to go upwards of 30 or 40 hours per week during more challenging topics.

Simply said, there is no substitute for long, hard hours.

Consider this time to be an investment in yourself, and know that for every hour you spend, you are guaranteeing yourself a better opportunity after graduation.

4. Patience Makes Perfect...

Tied to the previous suggestion is a second piece, often forgotten. Not only will learning to code require many hours, it will also require many months (and frankly years) to master. Don't rush success!

For almost all of you, the first couple months or so of the program will be particularly challenging. In fact, for many of you, this period will be one in which you doubt your eventual success through the program. Resist the urge to give in or to become hopeless.

Realize that learning this craft will require consistent effort, which will iteratively build your skills and understanding. What will seem challenging, confusing, and distressing in Week 3—will seem completely second nature by Week 24. *Intensity is no substitute for consistency.*

5. Not a Spectator Sport...

As every musician, painter, or craftsperson will tell you, one cannot learn a skill by simply reading, watching, or following along. Instead, it requires hours of deliberate practice. In the same way, coding, just as any other skill, will require you to step away from the spectator stands and to enter the fray yourself.

Know this now: going to class, watching tutorials, reading articles, or participating in other passive activities will only get you so far. You must spend significant hours actually *coding* to succeed in this bootcamp. As a benchmark, consider spending 70–80% of your outside class time writing and reviewing code. Only the remaining 20% is for other

passive activities.

But I don't feel ready!

Often, we've seen that students who struggle the most are the ones most intent on *learning everything* before they start coding themselves. Unfortunately, this is a recipe for stagnation. You will not be successful in this program unless you force yourself to work through confusion: writing code, making errors, chasing them down, and learning from your mistakes.

This bootcamp is a mudslide. There's just no way to avoid getting your hands dirty.

6. Crumple the Paper Tiger...

Am I doing this right...?

The five most hated words of every development instructor.

These words aren't hated because your instructor is unwilling or is unable to help. Rather, they're hated because they suggest that a student is afraid to *try* something on their own. As you enter the classroom, learn to abolish this phrase from your vocabulary.

Instead, learn to try things on your own, to test what you can, and to do a bit of debugging first. *Then* turn to your instructor and say, "I tried such and such. It didn't work so I tried such and such. But it still didn't work. What should I try now?"

Notice the change in tone. Instead of timidly asking for assurance, give yourself permission to *just try*.

The best way to fail is on your own terms.

7. Find a Squad...

For whatever reason, Hollywood has created an image that every coder is an isolated loner, programming in a cubicle. Don't let this be you!

While in this program, make every effort to find friends, to form study groups, and to work together in and out of class. Sometimes, the fastest way to overcome a bug or to understand a challenging concept is to have another pair of eyes or to have another perspective. Pair programming is a real thing for a reason!

8. Master the Art of Google Fu...

One of the greatest surprises to students entering the program is the amount of *Googling* they are asked to do. This isn't because your teacher is lazy, or the curriculum is unplanned - far from it!

As every professional developer will tell you, coding isn't about *memorization*. Instead, it's about curating bits and pieces of knowledge, scattered across the web and accumulated in various documentations, forums, and QA websites.

To become a good developer requires an ability to work through problems and to quickly *research* solutions ascertained by past developers. The field is constantly changing, and new tools are always on the horizon. As you will find, every good developer is a great friend of Google (and Stack Overflow).

For some really cool tips on how to become better at Google Fu, check out this article!

9. Plan Often...

As you will learn in the pseudo coding module, the best first step in *every*coding challenge is to formulate a plan. Break down the complex task in front of you into discrete, bite-size coding challenges. Once you have a plan, write it out and always refer back to it.

Fundamentally, every task in code can—and should—be broken into smaller tasks. Don't try to bite off everything at once! You'll get lost in your own mind games.

10. Fixing Things Takes Time...

One of the most frustrating aspects to new students of software development is the sheer amount of time it takes to troubleshoot (or debug) issues in code. At times, it might even feel like *fixing* an issue is taking 3–4 times as long as conceiving the original solution.

Know in advance that this is completely normal.

For novices and experts alike, fixing code is often the most time-consuming task of all. Instead of seeing these spent hours as a *distraction*, learn to see them as a critical part of the learning process. Each bug you pursue is a lengthy lesson that adds to your arsenal of understanding.

11. Self-Care is Key...

While we've probably traumatized you with all the talk of challenges, of difficulties, and of effort, we do *want*you to take care of yourself. Throughout the program, be sure to sleep, to exercise, and to eat nutritional meals. These moments of self-care are extremely important for your mind to be healthy. Taking breaks is encouraged!

In fact, you will find that that some of the best problem-solving happens during breaks. "Sleeping on a problem" is often a very real solution to your most challenging coding issues. Try to walk into class each day, ready and refreshed for new learning. We want happy, energized people in our classrooms. Not dead robots.

12. Be a stellar student

As a student, we'll encourage you to collaborate with your classmates, and participate in study groups in and out of the classroom. We'll ask you to look for code examples in the curriculum and online as you're learning. This is good practice for a developer as you'll do the same on the job.

Just make sure you're doing your own work and submitting applications you can be proud of. If you feel like you are borrowing too much, reach out to your SSM for guidance.

There's value in the work you're doing, it translates to hard skills, and you truly get out what you put into this program.

13. You Can Do This!

This last piece of advice is the most important. Remind yourself each day that you *can* do this. We've seen through many classrooms, students from all backgrounds, experiences, and personal situations persevere and succeed through this program.

During the tough times, dig deep into your own personal motivation and remind yourself why you entered the program. Let this be the fuel that you use to keep on.

You have *everything* it takes to learn this craft and to gain the opportunities that come with it. It may feel challenging at times but remind yourself of past challenges you've overcome. Your future self will thank you for all that you endured.

Video (Recommended)

- Grit: The Power of Passion and Perseverance
- James Clear: Successful Habits
- Monica Szabo: Student Inspiration

Assignment (Required):

- My Greatest Challenge

Additional Reading:

- Learning Programming: How I Overcame Initial Resistance
- Learning to Program at Age 30

Copyright