THE BOOTCAMPER'S GUIDE TO THE GALAXY

(and other things)

This page was intentionally left blank.

FOREWORD

Hello and welcome fellow bootcamper. I know these past few days have been tough on you. It's been a lot to take in. All the new faces you've seen; all the new lessons you've learnt - it's been overwhelming, but I'm sure you didn't know you were this tough. I know you've contemplated giving up on multiple occasions, because hell, I have too. Yet here you are: with more fire inside of you than ever before.

By now you've probably learnt that everything you've experienced is to toughen you up, shape you and then ship you out into the big wide world, with more than just theoretical knowledge and a fighting chance - with a better chance at success than most. What I have taken from this bootcamp thus far, is that we are trying to create a family - one that lives together, cries together, laughs together and most of all, grows together.

In the spirit of We Think Code, I also believe that sharing all that we know is key to not only improving the knowledge of someone else, but also that of ourselves. Only in teaching what we know, does the information that we have been taught become solidified within the chasms of our minds. With that being said, I wish you the best of luck. May your code be strong and your mind be clear. May the force be with you.

PS: Like you, I am an imperfect student. I am no pro at beating the system (yet), These guidelines are just that, guidelines. By reading or implementing them, you take full responsibility for your code. You also agree that I cannot be held liable for any errors or misinterpretations of these guidelines and that they are provided to the best of my knowledge and with the best of intentions: to help you succeed. If you feel that I can improve this guide, I'd love to hear from you. Your input is important to me. Pop me a mail at gani.kyle@rocketmail.com) or see me at class :)

~ Kyle Gani

This page was intentionally left blank.

TOPICS OF DISCUSSION

INDEX WILL GO HERE ONCE THE GUIDE HAS BEEN COMPLETED

CHAPTER 1:

THE WEEKLY EXAM

I won't lie to you: Big brother expects you to fail the first exam and do poorly on the second. It doesn't mean that we have to lay down and take it. As bootcampers we expect more from ourselves. We cannot expect that we'd never fall down, though. It's all about getting back up (and quickly), because in an environment as fast-paced as this, we can't afford the time to lay there, licking our wounds.

I won't give you any answers or what kind of questions you can expect in the exam. I believe you have your own mile to walk. That can only happen once you've failed dismally, picked yourself up again and reached the finish line - tatty shoes in hand. I will however, assist you by providing the methods that I've used successfully thus far with the help of my friends, to ensure that you stand a fighting chance. I believe we all deserve a fighting chance.

Preparing yourself for the exam

Exams, as with everything else here at bootcamp, are like nothing you've come to know. They're longer, more difficult and much trickier than the one's we've been completing all our lives. As a result, they require something special.

one of the biggest, challenges I've faced at the bootcamp is having a clear mind. between all the deadlines, peer corrections, community service, friendly distractions and our lives outside of campus, we hardly ever think about the state our minds are in. That is a HUGE problem - one that needs to be resolved quickly.

A very clever trick I have found to save myself from drowning in my own thoughts isn't a

breathing exercise. It isn't a morning dance routine (no judgement if that's your thing). Oddly enough, it is something as simple as talking to myself. Mentally conditioning yourself is superbly powerful and when done correctly, it can be very liberating.

The exams are the only time at bootcamp where you are completely alone. You have no peers to correct you and no-one to help you - except for if your computer explodes. As such, you have to physically change modes; from the outer more social person, to being completely in your head, by yourself. we can't expect to have the same mentality for both of those situations.

Before an exam, I physically will myself into incognito mode. When I walk through the door I imagine that a barrier strips me of all my problems and keeps them locked away in a vault until I return. Our external problems can do nothing for us, so we might as well leave them at the door.

I take my seat, close my eyes and do something quite weird. I imagine myself in front of myself. "How present are you?", I'd ask myself. "How in-this-moment are you?" If my answer isn't between a 7 and a 10, I would mentally psych myself up to that level regardless of how tired I am, how stressed I am and how little I know. It makes a world of a difference. It helps me focus, it helps declutter my thoughts, and mostly, it helps me fully experience the moment. It puts me in the zone. Trust me, at one point or another, you're going to need to be in the zone.

Try it, see if it helps.

Now, I'm sure you've noticed that I haven't been speaking much about studying your work. You'll be right to assume that it is something you hardly even need to do if you play your cards right. I'm not advocating that you neglect going through your work. I am however stating that if you do as many peer corrections as possible, ask enough questions and build your skill through practice; that retrieving code in the depths of your mental hard drive will come as naturally to you as forming sentences. Code is a language, after all. You don't reach for a dictionary every time you want to have a conversation now, do you? The only way to improve your language, is to eat, sleep and breathe it - just like your code.

The environment

Rule 1: Dont be intimidated by the staff. It is their duty to make it grueling for you. They have to whither out the weak for the sake of the initiative. Don't be weak.

Don't take it personally either. You know and they know that you are born to code. The tests proved that. You wouldn't be here if you weren't. Just have faith in yourself and your peers. Shut out everything else. In a high pressure environment, the best thing you can do is keep your cool.

Another thing: make sure you know the exam rules. My cellphone rang in the second exam and I was sentenced to four hours of community service (I was lucky to not have been kicked out of the exam). The evidence is all over Youtube. The community service was fun, but being locked out of my computer - unable to work for four long hours - wasn't. Be smart.

The technical stuff

Right off the bat, you'll have to log into a user account that isn't your own for examination purposes. At the time of writing this guide the username is *exam* and the password is *exam*.

You'll notice that it is quite different to what you're used to. You don't have access to the internet, nor do you have access to any other applications, except for terminal, iTerm, your recycle bin and preferences.

I always start off by setting my mouse preferences. I'd activate the right mouse click functionality within the preferences app (Don't ask me why they leave it disabled - it must be Geff's idea). I'd also set my scroll direction. As I've been a windows guy all my life, I just find it easier (I like up to be up and down to be down).

I'd then go about launching the terminal to start my exam session. Seeing as we aren't in our usual user profile, we have to set up our kerberos tickets so that the system understands who it's grading. This could be done by typing the following:

kinit [username]

Obviously, the word username will be replaced with your own username supplied to you by the French system. You will be prompted for your password (It won't show anything while you type, for security reasons, so type your password as best as you can without being able to see what's happening (once again, it'll be the one supplied by the French system)). Upon successful completion, you will then be greeted by your standard computer number that we normally see in our terminals. Type the following to launch your exam session:

examshell

A list of all your exams will be made available. Follow the prompts to get started. You will notice that a script will run, cloning the exam git repository into your user profile, usually to a folder called *rendu* (we'll call this the git directory). Once that is completed, information about your particular assessment will be provided to you.

At this point, I'd launch 2 iTerm windows, by first clicking iTerm to launch one window, and then right clicking the icon in the task bar and selecting *new iTerm Window,* to launch another. One of the windows is where I will code all my files, and another is what I will use to refer to the instructional text files. Feel free to launch as many as you see fit.

We should now have 3 windows open: examshell, a coding window, and an instructional text window.

If you navigate into your home directory using your coding window, by entering the following command:

cd

You will notice a few folders. The only ones you really have to take note of is *rendu* (or your git directory) and *subjects*. your git directory is where you will house all your code, and *subjects* is where all the instructional information for your assessments will be housed.

Navigate into the *subjects* folder where you will see your projects (The current project can be seen by going back to our terminal, which we've used to open examshell). Navigate into the assessments folder (e.g ft_rot976) and vim into the text file.

Pop over to your code window and create a subdirectory within your *git* directory as directed in the examshell. It should say something similar to, "your project should be located in rendu/

[assessment name]“ in bold green text (e.g. rendu/ft_rot976) . These are the individual folders we will be pushing to Git. Within this folder, create your files as indicated by the assessment text file.

Once your code is complete and you have tested it thoroughly, remove any unnecessary test files and executables that were created (unless instructed otherwise) and navigate into the parent directory by typing the following:

cd ..

Now we should be in our *git* directory (e.g *rendu*). If we're ready to have the assement graded, enter the following code while in your git directory:

git add [Assessment folder name] // e.g git add ft_rot976
git commit -m “[assessment name]“ // e.g git commit -m “ft_rot976“
git push

Move over to your examshell terminal window and type in the following:

grademe

Follow the prompts, the screen should now do a few things, and you should see either success or failure.

A few tips

Ensure that your files are in the correct folders and make use of the exact naming structure provided to you by the examshell. You will FAIL if files are in folders unrecognized by the grading system.

As far as I know, norm is not as strict as in our daily exercises (If it is being used at all). For the sake of continuity though, stick to it as much as you can.

Test your files to see if they work. Create the necessary files and functions to test your programs and functions. In your assessment text files, they have a few examples. See if you can recreate them.

If they ask you for a program, include your main function, additional functions, any allowed functions and any headers you might need. If they ask you for a function, only include that function, any allowed functions and the dependent headers for those functions!

If you think your code will be longer than 25 lines per function, you are allowed to split them into 5 separate functions (as long as your reference, dereference and call them correctly).

Ensure you get enough rest before the exam. Not being able to concentrate is detrimental to your exam session - especially after the 24-hour challenge.

Don't rush. I know that it's a shit-tonne you have to get through, but focus on each project at hand. Completely understand what you have to do. It's okay if you're the last one in the exam room - as long as your code works!

Don’t be too hard on yourself if you fail. Exams are meant to test your limits. Learn from them just as much as you would in your daily exercises. Ask yourself what you can do better, discuss it with your pod or in your next peer correction… and then do better the next time around.