

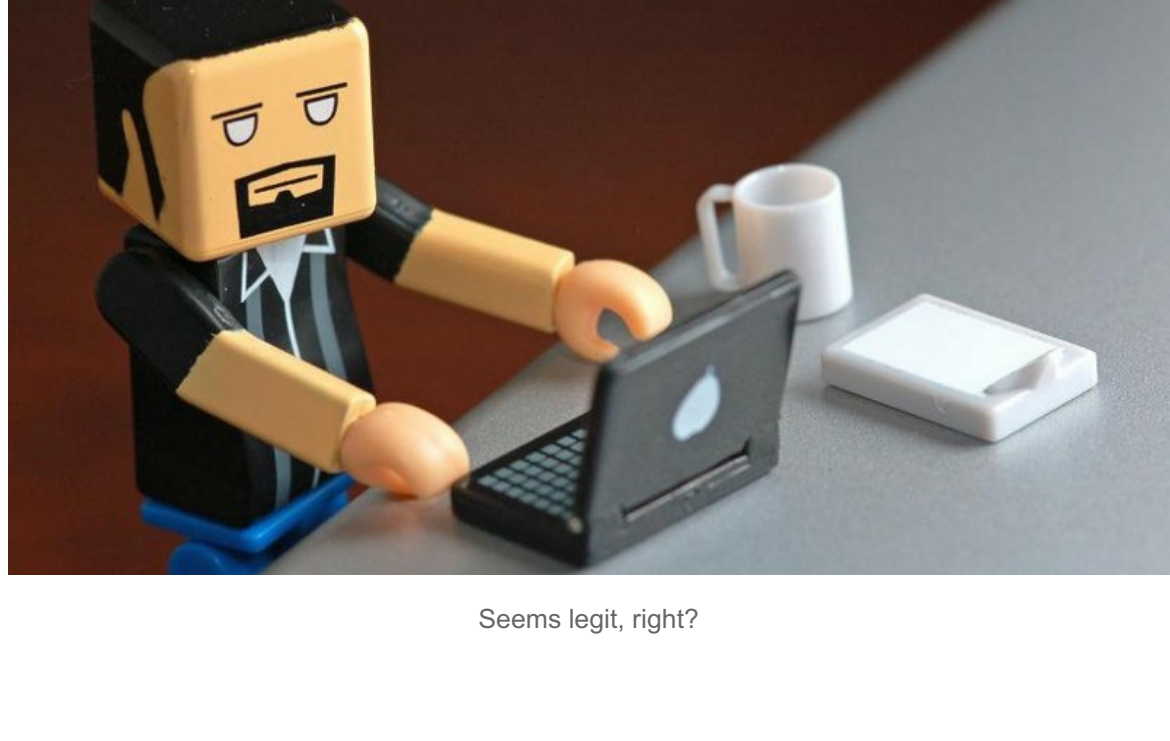


shakycode [Follow](#)

intj | autodidact | coder | sec | coffee | linux | ruby | rails | lulz | devops |

yesterday · 8 min read

## The Story Of The Fraudulent Coder



Seems legit, right?

Before I go off on a rant I have to stay on this theme...

### Programming is fucking hard!

There are no shortcuts to becoming a great coder. It's all about intelligence, humility, hard work, making mistakes, learning from those mistakes, and comprehending the material you are presented with.

But some people think there are shortcuts and trust me, I've tried to find them all. The only shortcut I could find which is totally unethical and lame is that of being a fraud.

This article is about the fraudulent coder and my experience trying to steer this particular individual in the right direction.

So it started a couple of years ago when I met a bright 30-something year old gentleman, for the sake of anonymity let's call him, "Bryan". Bryan had an avid interest in web development and had been playing with Ruby on Rails for about a year. He even went to one of those crazy-expensive coding bootcamps (ugh) to turn him from zero to hero in 12 weeks. He was a contract manager by trade and had 12 years in the business but he was desperately trying to break into development. Being a natural mentor and always wanting to share what I know I took him under my wing and started pairing with him at least twice a week.

We would work on his apps (mostly blog and todo apps) and slowly but surely he started to learn the basics. For being in a 12-week bootcamp I was shocked by how little Ruby or Rails he actually knew. I encouraged him to really study the Ruby language deeply as Ruby is the core of Rails and without it he wouldn't get very far by relying solely on all the magic that Rails gives us. He promised to work on pure Ruby on his downtime with me encouraging him even further by sending him copies of Ruby books that I really liked and helped me learn. I thought I was being a good mentor, boy was I mistaken.

#### Redflag #1

I was following him on github so I often checked out his repos and made suggestions on how things could be improved or perhaps discuss design choices. One day I woke up and checked his github account and found 5 new repos almost fully completed. Each repo had 2-3 commits which were large in size. It's as if he had copied an app and committed it to git in big chunks. When I asked him about these 5 new apps that sprung up over a day his response was, "I was working through 5 different tutorials and wanted to fill up my repo in 24 hours".

I asked him, "did you cut/paste the application code or did you actually follow along and type it out yourself to gain a better understanding?". He responded, "This is all too simple, so I just cut/paste the code into my repos and did massive commits. This will look great to employers!". He seemed enthused but at the same time I needed to put him in check.

While many of us will cut/paste solutions that we find on the Internet the difference between a newbie versus a veteran is that the newbie introduces code that he/she has no idea how it works, just that someone else has it working. They worry about the "how" much later. The veteran will cut/paste the solution, read through it, understand what it does, and refactor for his/her specific use case. Cut/paste coding is discouraged, but it's not the end of the world if you borrow someone else's code as long as you give credit where credit is due.

I explained to him that employers will ask him about the code and when he goes to explain how it works and why he chose to follow that pattern that he would come up short and be exposed as incompetent. We literally argued back and forth on this for a while until he finally said, "I'll leave these repos up and study what I built so that I understand it". I went through several pairing sessions trying to get him to understand but for some reason he was insistent that he didn't need to understand it so long as he has a vague idea of how it works and that he could reference his work if need be. But this wasn't his work! I felt very discouraged by his behavior. If he was trying to land a real junior developer job, this was not the way to go about it. Employers want to see *YOUR* actual code, not some crap you cut/paste from a tutorial just to give you a false sense of credibility.

#### Redflag #2

One day Bryan reached out to me on Slack and asked him if I could help him work through a Ruby exercise. It was a basic fizzbuzz challenge that most employers will throw at you to see if you have any clue. I ended up writing the fizzbuzz exercise for him and explained to him how it worked. His response, "Great I'll turn this in first thing this evening!". My response, "Woah, wait. This is a code test for a job and you just had me write it for you?". He quickly signed off of slack.

I didn't hear from Bryan for a few days and he finally came back on Slack and told me he had a final interview for a Junior Rails Developer position with a startup. Not trying to be a dick, I congratulated him on his progress and wished him luck. I didn't mention the code test or the fact that I solved it for him. I just kind of let it go.

#### Redflag #3

Later that week I saw a new repo on his github account that was named "capstone\_test". It was a fully fledged CMS app that was only 18 hours old. There's no way in hell he could have built that in 18 hours unless he had help. Now mind you, I'm basing the time constraints off of his ability to read and write code. Since this was a public repo, I made some commit notes on where he could better improve his app, being the mentor of course. I left it at that.

He contacted me on Slack a few days later and thanked me for the suggestions and that he committed the changes and submitted it to his potential employer for review. When I asked him how he built such a complex CMS in 18 hours he told me that he used AirPair or some other mentoring service to help him build the app. I was seeing absolute red and was furious with him. First he had me help him pass his initial code challenge, then he paid (\$100/hr) to AirPair or whatever service for help building his final coding assessment which was being submitted to the employer.

#### Redflag #4

I really don't pay attention to my github account as far as traffic for each repo and who cloned my stuff as I really don't care. But I just happened to check 3 of my production-ready apps and I saw a unique cloner for each one. I thought that was odd, but didn't pay much attention to it. Later that evening I looked on Bryan's github account and saw my repos on his account. They were not forked, but they were actually put up as if he wrote them. There were even a bunch of shitty commit messages all bunched together. No way he could write 3 apps in a day. While my code was open sourced and I didn't care who used it, I wanted it to be used for a good purpose, not for facilitating fraud.

### Bryan gets the job!

LinkedIn notified me that Bryan is now "Software engineer at Acme Company" so I congratulated him on the job and wished him luck. Bryan got real quiet for about 2 weeks until he started his first day on the job. He began to ask me all sorts of questions that were way over his head (one in particular had to do with websockets and how data streaming works and the best way to implement it in Rails). I sent him some links and asked him what he was working on. He was working on a realtime notification system at his new job and he admitted to not knowing what he was doing. I was furious. Here he is landing a good junior developer job and he was basically a fraud. And what even got me more upset was the fact that his salary was 20k above mine and yet he only had 6 months experience at home with zero professional experience.

I had to make a decision with Bryan and fast. So I decided to ignore him on Slack and just let him fall on his face. If my mentor found me doing fraudulent shit like this, it would have been 10x worse.

Blacklisted and booted permanently.

### Bryan Loses The Job

A month had passed without a word from Bryan. I kept an eye on his blog and he was publishing a coding tutorial every week. I felt that maybe he was learning but these articles were literally cut/paste solutions from Stack Overflow and the explanations were plagiarized as well. This was absolute fraud. One day Bryan came into Slack and asked if anyone knew of a company that was hiring. I asked him what happened to his new developer job. He told me that it wasn't a good fit for him and he's moving on. Little did he know that I knew someone at his company so I dug deeper for more information.

My inside source told me that they felt that Bryan was not even a junior developer and that his solutions were all from StackOverflow. Apparently they gave him 3 weeks to get his shit together or they were going to fire him. I divulged to my inside source that I wrote the fizzbuzz test for him and that he used AirPair for his capstone test. My inside source was furious and called the VP of engineering immediately.

Within 2 days Bryan was terminated from his position and back to doing contract work in his old industry.

### Lessons Learned

- You become a great programmer by putting in the work, learning from other's work, and not stealing other's work
- You learn from your mistakes and use them as growth catalysts
- You do not cut/paste code without giving some credit to the original author
- You don't bullshit your mentor and get away with it
- Plagiarism is frowned upon and is the easiest ticket to getting shitcanned

### Summary

After all of this went down Bryan tried to reach out to me for help finding a new job. I informed him that I could not work with someone I could not trust and that he was on his own from here on out. I never heard from Bryan after that day and to this day he's still cutting/pasting solutions and searching for a developer job.

#### Moral of the Story

DON'T BE BRYAN

- Shakycode ❤

Twitter: shakycode

Instagram: shakycode