Programming Is Not for Everybody

http://web.archive.org/web/20140711171601/http://symbo1ics.com/blog/?p=1615

By Robert Smith, on March 3rd, 2013

Recently, this video, provocatively titled What Schools Don't Teach, was posted, and has been an internet success. The central message of the film was "anyone and everyone can program", and it stars famous recognizable wealthy people asserting that point. (By the way, those people obtained their own wealth largely by hiring what they say they want.)

Also recently, my brother—who I affectionately describe as someone whose biggest dream is to be rich and will look into any scheme (albeit legitimate and legal) to get rich quick, whether they're shady entrepreneurial partnerships or plans to "beat the house" in blackjack or poker—came to me saying "I want to be like you, I want to program." (He too saw the video above, after having said that.)

I want to say that I think all of this is a hugely false message. Not everyone can program (beyond simple tasks), and more importantly to the eyes of prospective programmers, not everyone can make a career out of it.

The video starts off with all of the famous people, like Bill Gates or Mark Zuckerberg, saying "I started programming before I even hit puberty" followed by their "humble beginnings" writing very simple programs like small games. And then Drew Houston, the creator of Dropbox, pops up saying something relatable to a lot of people: "[programming is] not unlike playing an instrument or playing a sport" followed by NBA All-Star Chris Bosh, who "coded in college", very vaguely discussing his trials and tribulations with programming ("it is intimidating"), while not actually giving any evidence he struggled.

Nothing of what they are saying is inherently false. Yes, a curious kid can begin to learn the basics of programming (just as I did), and yes, learning to program is like learning anything else non-trivial in the world (like an instrument, sports, juggling, sewing, …). But then things begin to become very misleading.

Makinde Adeagbo, an "early Facebook engineer", comes on saying that a lot of coding people do is very simple. In a sort of twisted sense, this is true. But I liken that to saying a lot of stuff people do in basketball is also simple, like running across the court or throwing a ball. Running across the court is a necessary element in basketball, but it is certainly not what is sufficient to be successful at it. In fact, you can run across the court so perfectly every time, but be completely unsuccessful.

And then perhaps my favorite part: Bill Gates saying you need "addition, subtraction, that's about it." That is, that is the only prerequisite you need for coding.

Again, this is a sort of half-truth. It is true you do not need much mathematics to learn how to code. But the issue is, as you learn more and more how to code, you will find mathematics indispensable. In fact, I would say that you need at least a good grounding in algebra to be a successful coder, and more than that if you want to take advantage of better ways of looking at things. But ignoring the prerequisites, I think he's giving the wrong message entirely, approaching the situation with the bare minimum. What would have been more accurate is "you only need addition and subtraction, but you should be inclined to learn more math, because you'll need it to read anything beyond your Java in 7 Days book."

Finally, the video starts talking about what it's like to live the life of a coder. Extravagant offices, free food, lots of play, beautiful views, and of course, lots of money.

I found this almost painful to watch on behalf of other programmers, having personally been through and subsequently being hired by one of the places they film: Facebook.

It is true that all of those things exist, but it's definitely the exception, not the norm, for a programmer. Most offices come equipped with a coffee machine, and a fridge to put your own food, often crowded with other people's food that has been sitting there all week. They are in office buildings with gray cubicles, and they are fueled by stringent bureaucracy.

So, all I have to do is get hired at Facebook. They just told me it is easy. They are hiring tons and learning to code isn't hard, it's just intimidating.

Except it's not. Almost everything that this video purports to be true is negated by the very companies that are represented in that video. The Facebook interview process requires a whole lot more than "addition and subtraction", and isn't purely fun and games in terms of problems they give you to solve. One of their interview questions requires an understanding of calculus, another one requires the ability to reason about time and space complexities of highly recursive functions, and another requires knowledge of abstract tools that programmers use, such as regular languages and automata.

As I said, my brother told me he aspires to be able to do what I do, a very flattering gesture. He knows it makes a lot more money than being a dock worker at FedEx does, and it allows a bit more of a lavish lifestyle. The video also portrays the same thing. While what he said is flattering, I don't think he sees the reality of it. The reality is not being "wired in", hacking out this amazing multi-billion dollar product, shooting caffeine through your veins, and typing at an incomprehensible speed.

I sit here this weekend, droning away writing **and reading** code for a large code base I don't understand. I feel like if this task isn't completed by a certain time, my job will be on the line. I'm not programming fun games or interesting concrete things. In fact, if the layperson like my brother asked me what I am programming, I'd have no idea what to tell them. My answer would be so utterly uninspiring, that it probably wouldn't even register as being a normal part of programming. "It's a sort of thing that reads code and is able to deduce information about it and then lets other people use that information for their tools" I don't think this is the kind of programming my brother, or anyone, expects to be doing.

Do I like getting to work, at the latest, 10am, and coming home, at the earliest, 7pm? No. Programming jobs are often **not** “clock in, clock out” jobs. There is an underlying expectation to work longer hours to get things done. And it never stops, because the work is never ending. There’s always some feature to add or some bug to fix. There will always be a reason for you to have to stay an extra hour or two. Personally, I feel like my time is so short that I have no other choice than to stay up until 12 or 1am in order to give semblance of a balanced work/personal life. And this pattern isn’t common with just me, it’s true with many programmers. (Interestingly, what I end up doing in my personal time *is* coding, which perhaps is evidence of the sharp contrast between programming for personal pleasure and programming In The Real World™.)

I don’t want to disparage the idea of someone new starting to learn programming. It can be an empowering thing. But, for me, it’s not something I picked up and learned to do over a weekend, or two weekends, or 52 weekends. It is something I tried hard at, and ended up neglecting other things in life to be able to learn it. It is hard to tell my brother, or anyone, that the fruit of learning a little bit of programming is the way it might change the way you think—not the fame and fortune—and I don’t think this simple thing is a good enough motivation for most people.

Quite simply, programming is not for everyone. It is not an absolutely fun and delightful task as it’s portrayed to be in videos like the aforementioned or movies like “The Social Network”. It is rarely an invigorating social activity. A lot of time, it’s sitting in front of a computer screen, looking at a colorful text document, and thinking, and thinking, and typing, getting angry, and wondering why the hell you’re living a life sitting down.