

JUNE SCOBEE RODGERS

In *Collecting Responsibility*, I recount a moment of inspiration. I wrote about *Challenger* in terms of a postcard I'd lost. It was a critical breakthrough in my thinking. Later that night it was as if the basketball gods were blessing my idea because KPJ destroyed the Bucks, dropping 50 points. It is a good story, and it happened like I wrote it.

It wasn't until much later that it hit me; I'd already finished writing the book and realized something even more profound. It's an even better story. This work is a testament to genius, so I'll keep the embellishment down to a Texas minimum.

I was curious, I remembered *Challenger* well, one of the few moments in high-school I can see clearly through the cobwebs of time. I also remember who I was, what I was like, and there was some incongruity. How did my mom get me to a NASA nerd camp when I was fourteen?

After a full year of 6th grade, I'd left my nerd roots behind. I still read science fiction but wasn't hanging out with other nerds who did. Summer after 7th grade was the last run of my suburban rap/breakdance crew. Thankfully, this occurred before digital photography. Home videos of teenagers doing stupid stuff was for rich people. Summer after 8th grade, I was hanging with kids who listened to metal and hated school, friends I'd have through high school. I know I wasn't volunteering to go to a NASA summer camp.

My Uncle Clay, retired Air Force, had broken the news to me years before. One summer it was, *with your glasses you'll never be a fighter pilot*. Gave it to me raw, then told me how he found out he was colorblind. I think the story was he didn't even know until after he'd already signed up for the Air Force, hoping to become a pilot. The Air Force doctors told him. No jets for Uncle Clay, but he made a career out of military intelligence.

I'd left astronaut dreams behind in grade school, along with the nerds. I wasn't a computer geek. I didn't even like to play video games at home, though the arcade was cool place to hang out. I couldn't understand how I ended up at NASA camp at that age. I asked my mother.

"You got picked because you were good at math," my mom said.

"Oh really," I say, not sure why that would have made a difference.

"Did you have to pay?"

"Oh, no," said my mom. "You went because it was free."

And then I got it. I get along with my mom great now, but as a teenager if there was a way to get me far away from her and it was free, she was first in line. My mom could walk into a supermarket in the 1980s with a box of coupons and walk out with two carts of groceries and \$50 cash in rebates.

She'd camp out like it was for concert tickets, to get rid of me for a few weeks in the summer. I can see my teen-self would have made peace with the fact it was all the way in Clear Lake, far away from where anybody might know me where I lived in Cypress.

I had no interest in computers at all. As a kid, I'd wanted to be an astronaut, but as a teenager, no way.

I don't remember much about the camp. We did a tour, saw mission control. We looked through a window at the computer room. I saw my first hard drive at NASA, it looked like and

was the size of a washing machine. I remember the engineers were dudes in short-sleeve collared shirts who'd chain smoke and wore bad glasses.

I do remember learning to code on the Apple IIs. Only time I ever used Apple anything until I got an iPhone 25 years later. It wasn't even a real programming language, but this educational language called Logo that drew pictures on a screen. I don't really remember them teaching us much, but they must have. I don't remember if June was even part of the programming class, but she was part of our day.

While we were in some lecture scenario, I remember June talking about science fiction and ethics and the **Laws of Robotics**. I never read Asimov's robot novels, I'd gone straight to *Foundation*, since my Uncle Clay had told me was the best Asimov. I might have mouthed off something to that effect.

I remember meeting Colonel Scobee, it might have been on the last day of camp. I hope I was respectful. I remember thinking, *he looks like somebody's dad, not a badass pilot*.

He was.

Both.

The tragedy is what stands out in my memory, what happened after. It was only half a year between the camp and the accident, but in a teenager's timeline it felt like a decade.

There was a lot of expectation in the news and among Houston schoolteachers about a teacher in space. We were freshmen boys, I'm sure we mocked it. My youngest kid is that same age now. His disdain for teachers matches mine at that age, for the same reason. People are complex, and kids see things through their direct experience until they've experienced more. Raising teenagers sure does change your perspective on being a teenager.

I have no memory of knowing that Dick Scobee was commander of the teacher-in-space mission ahead of the tragedy. I bet my mom told me and I tuned it out. That January day, realizing what it meant after the accident, was sickening. That memory became a stake in my identity. How shallow we can really be.

I can't blame teenagers for their thoughts on teachers and authority. Look how long it took me to realize the true value of this gift from June Scobee.¹

Her gift was the greatest leverage in real terms anybody has ever given me. I raised a family and provided for them with this gift. I was 51 years old when I realized it. I was 14 when I went to that summer camp. 37 years. I have made an entire life out of what June taught me I could do in a free summer camp.

This is where the story diverges from what most teachers would want.

I had no interest in technology in high school. I read a lot and was smart and could figure things out. I didn't like being forced to learn what I thought was basic stuff. In retrospect, the skills I'd learned to that point, going to elementary and middle schools in Cy-Fair ISD prepared me very well for the life I've had. Long before the internet, I had the capacity to figure out what I needed to know and the skills to find the answers. I could read, I could write, and I could do math and basic algebra, and that has served me well though my life.

¹ A reliable Texan source has confirmed that the camp was created by June Scobee and sponsored by Texas A&M. Karma at work, a lifetime of Aggie jokes has finally caught up with me.

After 11th grade we found a loophole, and me and my friend dropped out of High School, took the GED and enrolled at University of Houston. Neither of us took to college any more than high school, but since we weren't 18 yet, it was an acceptable compromise for our parents. The next year, we both took off for Los Angeles. I pursued music and writing for a few years, doing a variety of odd jobs to support myself.

I knew what a PC was but never used one. In fact, rather than buy a computer, I spent a few hundred bucks on an integrated word-processor with a printer. I had zero interest in technology or engineering, I was writing about what I wanted to write about.

I had a bunch of food and retail jobs, delivery driver, bank teller, liquor store, gas station attendant. In 1992 I was working for a redneck in Waller delivering horse trailers, when my vagabond friend set me up with an air-conditioned job repairing laptop computers. I ordered parts and kept the operation moving along and learned how to put laptop computers together. The secret, I learned, was that if you put a laptop back together and have an even amount of screws leftover, you did something wrong.

That company was somehow connected to Tony in New Jersey, and it went under. My drummer's mom worked at Compaq Computer on the night shift assembly line and used some insider voodoo to get me a temp job under her general protection. We worked on a Printed Circuit Board (PCB) manufacturing line, we made huge motherboards for servers. It was an air-conditioned job, and I learned several roles rapidly, but none needed computer skills. Late at night, you had to stay awake or risk instant termination if you were spotted dozing by a supervisor. It was very difficult as the line was often stalled for hours at a time due to changeovers or supply chain issues. There was one lady who did stuff on the computer. She'd open reports and print them, then collate them manually and then print out another report, which she would give to the boss. You couldn't read books on the assembly line even if there was nothing to do. I quickly figured out the best thing to do was look busy when there was nothing to do or somebody would make me do something I didn't want to do.

Computers in those days came with a set of manuals, including a BASIC programming manual. Extra books that came with new systems were always around the line and were the one kind of reading material allowed on the floor besides specifications, work orders and schematics. I found that I could teach myself BASIC by figuring out how to collate all the files and print them out and save the lady some time each night. It probably took about a month.

Didn't go like I thought when I unveiled my innovation to the lady. Turns out she liked staying late to turn in the reports so she could get a half-hour overtime pay every day. Plus, complaining about staying late to the young people gave her a way to stay awake all night. My drummer's mom rescued me, but I was banished to another shift in a negotiated truce.

After a couple years of this, I found my way into an engineering team. I started doing tech stuff around the lab. I automated some stuff with batch files. An opportunity came up for a programming assignment in C, and my brother had a book about C, so I told them I was qualified. Turns out, I was, thanks to June Scobee.

From that path on, my career through digital logic has been smooth sailing. The basic skill I learned back in 1985 was all I needed. I call that integrative thinking, the ability to break down problems into smaller pieces, and solve the pieces. It's a common skill, used in all industries, with or without code.

The more you do it, the better you get at it. The real power is your ability to use code as a tool to apply even greater leverage. That's just looking at real world problems the same way as you solve code problems. It doesn't take four years of high school and multiple years in university to learn those skills. It takes identification of natural potential and a little directed on the job training. Generations raised on digital logic are naturally more adaptive to integrative thinking because that is how software works.

I remember I'd been programming in C++ for about two years when I finally got some real training in a class we took at Compaq. My mentor, who was a C programmer only, but light years beyond me in engineering terms, was sitting next to me struggling with some examples in C++ which was new to him. I remember him watching me do some trick on the command line in Windows that was illegal in Unix and him pointing that out and I felt so smug.

That memory is truth. How leveraged by the past we are. My bag of tricks in programming was like a cheat code in a game to him. He'd had to solve all these puzzles himself, through incredible hours of focused thought. Every time I ran into a difficult programming problem – even in 1998 – all I had to do was write the problem clearly in English and type it into a search engine and I'd find at least a hundred ways somebody had solved the same problem.

Leverage is the prime component of any engineering task. Digital leverage is the most powerful tool man has ever created. That is because all engineering is digital engineering. Good engineers in every field are the ones that use software as a tool. *Engineers are responsible for the tools they use.* That's the most valuable engineering lesson. The sooner engineers understand their basic toolset, the more effective engineers they will be. All industries are potential innovation frontiers for young people trained to use digital tools.

Coding isn't for everybody.

This ability is not the same as being good at math. The math we use in integrative thinking is very simple.

I don't know how June Scobee identified that I was a good candidate to learn how to code.

I suspect if you ran the same studies to identify kids who'd be good engineers in the 1980's on kids from the 2020's, it would result in a much broader candidate pool. How could it not? We give very young children digital devices to play games with, sometimes before they can walk or talk.

It could be some kids are good at games because they have an innate talent, like some kids can learn to juggle. I doubt it. Video games are designed by people who like to play games, so there's a level of insight you can gain from that.

There are some kids who will always beat a game, given enough time. You can study those kids, and by studying the choices they make, figure out the deductive logic they are using.

Some will follow logical deductive paths. Essentially, they reverse engineer the algorithms the game designers used.

Others might just use monkey-with darts and make a map of what works, either mentally or by recording results.

Both approaches are good strategies to find a solution, and both are demonstrations of integrative ability. The map strategy requires many more iterations. In digital engineering that is natural leverage – something computers can do very quickly. It's easy to design experiments to iterate through a set of possibilities, that's the leverage of Moore's Law.

Digital games usually make you go through a lot of work to keep experimenting with a map strategy, so kids with this ability may tend to give up more easily than those who use a more deductive path.

Other kids might just be lucky. Some will have extra-leverage – they used a cheat code. That’s super integrative leverage – exactly what I do when I Google a solution.

This approach is a fair way to study and find kids that could be productive engineers.

You find these kids, wherever they are, teach them to code in a summer camp, and they will have a valuable and marketable skill that they can use directly in almost any endeavor they choose. More directly, you make sure these kids have opportunity to develop that skill in direct and equitable ways.

In Academic settings most kid’s prime motivation is to get it over with. Applied learning is often driven by the desire to increase earnings, a real motivator. Having a chance at a high paid profession is a motivation that leverages all parties involved. Dedication is required, but knowing you have the basic skills is all it takes to get started.

Learning to code directly changed my life. Thanks, Mom, for banishing me to the free NASA nerd camp.

I remember Dick Scobee as just an ordinary guy with an extraordinarily cool job.

June Scobee didn’t come across as ordinary. The correct Texas term for June is “sparkplug.” She wasn’t an ordinary mom and schoolteacher. She was like a live action version of a Jetson’s supermom, whose kid would be expected to grow up become a general in the Space Cadets. She expected something of her students.

Thank you, June Scobee Rodgers.

I wonder what the success rate of those kids June taught that summer?

After the accident, June and the families of the crew founded Challenger Center. Challenger Center’s mission is a natural extension of STS-51-L’s mission – The Teacher in Space Project.² Challenger Center inspires and educates students in STEM through hands-on lessons that simulate space missions. In the 35 years since Challenger Center opened, they’ve taught millions of students and enabled hundreds of thousands of teachers.

What’s the value of June’s contribution to our civilization? It’s very valuable, but immeasurable with property. You can’t put a dollar value on dreams. They have intrinsic value.

If you read *Silver Linings*, June’s book, you know she had to fight just to study what she was interested in. She gets her revenge by forcing the next generation – me – to learn something I was good at but had absolutely zero interest in. How ironic. And recursive. And Texan. And wonderful.

Genius.

My mission is to give every child on the planet the same opportunity that June gave me. To learn to code for free. I don’t know how I’m going to do it. I know who I’m going to ask for help.

Thank you, June.

Ad astra!

² https://en.wikipedia.org/wiki/Teacher_in_Space_Project