Like the vast majority of my Class of 1972 colleagues, I came to Penn after I had had a fairly easy time of things in high school. Yes, I had to work, especially at higher level math classes, but my high school transcript accurately demonstrated general competence. I traveled the 2500 miles from my native Southern California to Philadelphia and was full of confidence that I could handle whatever was thrown at me. My freshman academic year at Penn ended up being mostly review. I was able to complete the year with a decent GPA despite attending to Naval ROTC and the fraternity to which I had pledged.

Basking in the confidence of my nearly 19 years on the planet, in April 1969, I considered how I would comply with the Naval ROTC requirement at Penn that midshipmen complete a computer science class. There were two options, Computer Science through the Engineering School (Fortran Programming) or Computer Science at Wharton (COBOL, I think). Without figuratively blinking an eye, I told myself, “I am good at math; I can take computer science with the engineers.” After all, I was the son of an engineer and I considered myself a math/science person. I was confident that I did not need to take the course pass/fail, either. So my Fall 1969 semester course roster included Engineering Science 110 (ES110).

For those of you who missed the dubious joys of taking computer science at Penn in the late 1960s (and probably the early 1970s, as well), there were two computing centers at Penn. The engineers and hard-core computer geeks used the one at 34th and Market Sts., the University City Science Center. The other computing center was in the bowels of the Wharton School in old Dietrich Hall. The computer at the Science Center was an imposingly large mainframe that was kept in an air-conditioned environment. Think HAL in the movie 2001. There were no video monitors and no keyboards. The computer was programmed by a series of computer punch cards, each of which contained one line of code.

ES110 consisted of a series of what I found to be rather difficult mathematical problems. The goal was to program the computer using Fortran to solve the problems. The first task was to understand the mathematical problem being posed. The second task, once the problem was understood, was to figure out the steps that would be needed for the computer to solve the problem without any shortcuts. The third task was to write the code using the Fortran computer language. The fourth task was to transcribe the code to data cards on the keypunch terminals. The fifth task was to run the program on the computer.
The fourth task was no piece of cake. The data cards were 7-3/8 inches wide by 3-1/4 inches high by .007 inches thick tan cardboard. The blank data cards were transformed into computer instructions on the keypunch machine, a combination typewriter and paper puncher. The code was printed in ink on the top of each card in capital letters (Lower case letters were not used then.) and the cards were punched in a pattern that conformed to the typewritten code at the top of the card. It was important to look at the cards to make sure that there were no typos because, if there was one typographical error on a single card, the program would not compile.

After the cards were punched, the program needed to be put in sequence. Then, one had to hike from the keypunch room at the Moore School to the computer center. The program was “signed in” and one had to wait one’s turn to see what would happen. Depending upon the traffic at the center, the wait could be from 5 minutes to over an hour. The result of running the program was printed on one of several large, noisy dot matrix printers on 11-1/2 x 17 inch green-bar paper. I don’t remember how the running of the programs was communicated, but the green-bar paper output and the punch cards were placed in cubby holes to be retrieved after the programs ran, or in my case, mostly didn’t run. One of the first things that I learned, despite having taken typing in high school (in which class I was mostly humbled by the secretaries-to-be), was that I was neither a good typist nor a good proofreader of Fortran code. On multiple occasions, I received a green bar printout with a line of code followed by the dreaded asterisked phrase, **SYNTAX ERROR**. Often this was due to a typo. Other times, I puzzled over the printout, trying to figure out why my program had failed to compile. Whenever a syntax error was noted, it was back to the keypunches to correct that error and then back to the computer center to wait for the program to run again. If I was lucky, the program would compile and would spit out some data. The data, however, might still be incorrect. If the data was wrong, then it was back to step one to look at the problem again. Unfortunately, this process turned out to be my most common recursive loop.

About four weeks into the course, long after drop and add, I concluded that I was not very good at any of the tasks associated with the course. The keypunching and proofreading were constant annoyances. In addition, the math problems progressively became more difficult. Though I had taken calculus in both high school and at Penn, the math problems mostly stumped me. Lastly, I was a terrible programmer in Fortran. The language was highly structured and completely literal—I was neither. I worked hard at the course; I spent hours trying to get the programs to do what I wanted them to do, but I repeatedly failed.

I had made my engineering science professor aware of my difficulties with his course on several occasions. He tried his best to encourage me and some of the other students who were struggling. At some of those meetings, I thought that I could read a slight touch of exasperation, but, to his credit, Professor Bruckner remained patient.

Near the end of the semester, on December 1, 1969, the first “draft lottery” was held. My fraternity brothers and I gathered around the television set in the chapter room to view an important event in many of our lives. My birthdate was selected as number 277, virtually ensuring that neither I nor anyone with my birthdate would be drafted for military service. I had spent the summer of 1969 not at Woodstock, but rather as a Midshipman Third Class aboard the USS New Jersey. I learned a great deal from the cruise, including that I was not a particularly
good fit for the military. My questioning nature was not an asset there (curiously enough). Those lessons, however, are for a different day and different essay. My good fortune in the draft lottery allowed me to muster out of the U.S. Navy, which I did honorably before the Spring semester in 1970. I still had one legacy from military service to address, however, ES110.

In mid-December, Professor Bruckner handed out the ES110 final, a take-home examination that consisted of some short programming questions and then, a longer problem that required a program to be written and run. After much effort, I was able to get my program to compile, but the program did not solve the problem. I ran out of time to complete the assignment, so I handed it in as it was and awaited my fate. The worst part about my failure was that I had to stew about it before it became official. Some days after I handed in the final, I received a telephone call from the professor asking me to meet him in his office.

I walked to the meeting with dread, and the winter landscape intensified my negative feelings. All the leaves were brown (at least those that were left dangling from the deciduous trees on Locust Walk) and the sky was gray. I met Professor Bruckner at the agreed-upon time in his small office in the Moore School. He looked at me and asked, “Mr. Myers, do you intend to take any more computer science courses while you are at Penn?” Without a pause, I exclaimed, “You must be joking!” He said, “Well, I am happy to hear you say that. I know that you worked hard at this class, but I really think you should find a different major at Penn. You got a 50 on the final, but I am going to pass you because I know that you tried to do the work.” That mercy D was a relief and a welcome Christmas gift.

Flash forward: I became a trial lawyer. I prosecuted criminal cases for 8-1/2 years in the Philadelphia County DA’s Office and then tried cases for 1-1/2 years at a small civil firm. In 1985, my law partners and I established our own firm. Some of you may recall that 1985 was billed as the “year of the [local area computer] network.” Guess who was (and still is) the tech person at our firm? Me. I was older and more mature, and not afraid to say “I don’t know.” I was motivated to solve “real world” problems. We paid a consultant for help. I even learned to program a computer (in BASIC, not Fortran). People both at the firm and outside the firm asked me some of their information technology questions. What irony!

For a number of years after we started our firm, I kept my Winter 1969 report card with the D in computer science in a secret pocket in my briefcase. Perhaps I felt a need to hold my ego in check or perhaps I simply needed to remind myself that it was all right to ask for help. I have worn out a number of briefcases since then and I don’t know what became of that report card. As the years have passed, though, I have realized how important this particular lesson in facing up to failure was. I became more able to appreciate the skills in others that I lacked. Best of all, when one of my sons said to me, “Dad, I have tried hard to do this, but I still don’t get it,” I could honestly reply, “I understand.”