How My Computer Scholar Career Started

At James Madison HS, Brooklyn, New York, year 1998, a group initiated possibly by the man named Mr Bosini, or math department went around recruiting students with high math grades and are in the honor math classes. Everyone I talked to, including my guidance counselor, advised me that I would be good at computers and programming. I had no idea about programming or computers then. It was a nightmare once I was officially in the class. I always knew I was smart and I could learn anything, but that class was NOT effortless, but the class became a heavy burden. First programming language was programming in Basic. The teacher name was Mr. Sanderson. He was a good man now that I think about him, very learned scholar with thick glasses. However, he didn't baby students or had strict disciplined organization in the way he ran the class. The class was always filled with the teacher complaining about student's mischiefs, noises, and disorganized activities. Yet, it was like this in many classes anyway. Math I learned came to me naturally by logic and concepts. Most everything I learned came from reading the text books but not from teachers anyway. Yet, I couldn't grasp the information in programming statements, basic concepts about computers' hardware, languages, and all other fundamentals.

In addition, more than half the students had computers at home and they already mastered the BASIC programming language beforehand. I was no match for many students in that BASIC class and other classes like PASCAL classes. I tried reading the Textbook and realized I would only be able to remember these statements and concepts by doing them over and over. I had no computer at home and I had no idea about the computers. I was just good at math. I also realized that being good with algebra, trigonometry, geometry, pre-calculus and other math concepts had nothing to do programming. I passed the first BASIC class with grade 70, barely passed the second one with 65, and failed both Pascal classes. Nevertheless, I was exposed to real geniuses in the computer classes. Some kids were coding games and animation programs (computer codes out of their heads) like I am writing this article now. I can never forget a Russian guy name “Alex”. He was a true genius. He could code in BASIC like writing English. At the time many people in the world didn't even know how to turn on a computer or ever seen one. There were also evil students who were dismantling computers apart and try to steal drives and other components in the class. I can never forget that Alex the Russian genius and other evil geniuses. I felt completely defeated and dumb when I was exposed to them.

At college I went for an Applied Math BS, foundation math degree for all engineering disciplines. I was happy and content until I had to use computers again. The nightmare came back. First, it was a degree's requirement that I have to learn at least one programming language. I learned C++. I took this class very seriously and respectfully remembering how stupid I felt back in HS programming classes. I got an A with above 94 grades, but I had no idea with concept of compilers, RAMs, CPU, computer architecture, and much other fundamental knowledge about the computers then. I did well as a C++ programmer and I thought I would never have to deal with "the junk" that has nothing to do with math. As I progressed deeper into mathematical techniques, concepts, problems, experiments, and tools, I was required more and more to use computers. The Prof Edmond Rusjan believed in me and gave me a job or vouched for me to get a job as a software engineer at Applied Theory Corporation at Syracuse, NY. All I knew was C++, but he knew I was very smart and he thought that was all I needed to be. He was right! After I learned C++, I was willing to be like Alex back in HS to be a Master of the C++. However, the deeper I was exposed to the world of the computers, the more I have to know. There are web development concepts, web servers like Apache, PERL language, Oracle database, HTML, CGI concepts, JavaScript, C language, and etc. No one in the world cared much about BASIC anymore then. Then I realized that I would be wasting my time trying to be a master of any language because by the time I mastered it, that language would become obsolete. True software engineers only learn "nutshell" concepts and "code flow" (or syntax if you will) about any language. This is the only way to go. This is how true mathematics is too. Now I understand about what people told me, “I would be good with computers”. They should have told me this way: "you have the learning power and the ability to grasp, decipher, and remember complex ideas; therefore, problem solving discipline and organization skills to apply to be a good programmer or a software engineer is with you."

Both programmers and software engineers are like little children trying to put jigsaw puzzle pieces together. They see all kinds of junk pieces that they try to put together to make a picture. Programmers do that within specific language they mastered. They dive into all kinds of junks they can find in a huge junkyard to build a home or a car. The real skills for a good programmer are his ability to recall thousands of little junks he sees everywhere to go back and get them when he needs. A good programmer is a master of one junkyard. He knows where all the junks are in that particular junkyard. A good software engineer is a master of multiple junkyards. Real software engineers are more than just a programmer. They have the knowledge of many junkyards and can go get the junks they need to build a beautiful car or home. That ability to work with, recognize, and recall to solve problems with those junks is a true similarity between a programmer and a mathematician.

Poor little me suffer a lot from listening to people who never programmed, was not good at math or computers yet were trying to tell me what I would be good at. They were talking about a very farfetched concept that would be fulfilled only by lengthy time yet could be failed so miserably in so many ways. That a few hours I spent with Alex at HS watching him code in BASIC like a monster always motivated me and challenged me to be where I am at now...

First, a kid good at math then turned into “narrow-sighted” mathematician; then turned into a programmer in college; then turned into a software engineer; then turned into a researcher and a “scientist”. Now I see the whole world in a big picture and can apply scholastic concepts and techniques to everything and everyone around me. Still, I loved the word mathematics and to be called mathematician more than “engineer” or “scientist” of any kind in anyway. Unfortunately, no one cares or respects about the essence and values of the mathematics and mathematicians in business world and in job markets. Thus, I closed my eyes and lived with titles like software engineer, IT engineer, researcher, and scientist.

Lastly, I define the word mathematics or mathematician for everyone:

Mathematics are the techniques to simplify and authenticate truths. Lies, falsehoods, deceptions, doubts, and obscurities are the enemies.

Mathematician = enemy of the liars.