Elisha Parslow

CS 273 Data Structures

HW9 Programming in Ten Years

1.       Author Peter Norvig discusses deliberative practice to mean to try something over and over again, but trying the thing at slightly more difficult level than your current skill level so there is always a chance to learn something new or improve. When you set the bar high, you will become better at what you are practicing. The other key component of deliberative practice is to analyze your work, studying what was correct, and what still needs focus. In the realm of computer science, the best way to get better at understanding a language is through the concept of deliberative practice. As discussed in the article, reading a textbook isn’t enough. You have to really want to practice what you’re reading. Putting the information to the test is the best way to understand how a computer works in respect to the language you are learning. There are also differences in syntax and what a language will do for you behind the scenes. If you take the time to practice programming in each language you want to become efficient in, you will gain a better understanding of how to use it, what kind of programming it should be used for, and a better programmer as a whole.

2.       “Your first 10,000 photographs are your worst” is a quote that supports the saying that practice will make perfect. When it comes to programming, the beginning programs will be messy and nowhere near perfect. It takes practice and dedication to write beautiful code that you actually understand. The idea of 10,000 photographs can be translated into it can take 10,000 programs or ten years to become expert in the skill of beautiful and efficient programming. Some iconic expert people Norvig listed such as Mozart and the Beetles took years upon years to be at the level they were. It is the same with programming, and always relates back to deliberative practice. To get good at something, you have to start in a place where you’re not good.

3.       The first ingredient in the recipe for success in programming, to get interested really resonates with me and was crucial to my decision to study computer science. I was originally studying Heath Sciences and found that I wasn’t truly able to connect with it and didn’t feel like I was getting what I wanted out of the work. After sitting in on a computer science I class, my interest was sparked in understanding how to communicate with a computer. I found it very logical and understandable. Once I began learning what programming was, I was hooked on the way of learning computer science. I thrive in the trial and error and the conceptual and critical thinking that goes into building a functional program. I am slowly starting to realize what I can do in the world of computer science and my interest makes me excited for where it can take me. The second ingredient I find applicable to myself is working on projects with other programmers. Talking through code and really understanding what it does is a pivotal component to my learning curve. After I make the connections of how a program works on a screen with how to explain it to another person, my understanding grows exponentially and I get more and more excited about computer science.