CIS-2288 REFLECTION PAPER

Most server-side language such as PHP have always been staple development environments in my mind. I’ve experienced it outside of our current curriculum and I have to say that it’s always been easy to work with and documentation has been relatively straightforward. You can learn a language inside and out, including its syntax, and built-in libraries, but how do you learn the best way to utilize that knowledge and execute it in the best manner possible? Well, an obvious one is real world experience. Another is our course structure that includes single assignment work, working in groups, collaboration and research. Whichever method you choose, some value can be gained from the situation that would equate to a learned usage type. You also need to do a little trial and error with different concepts. Putting yourself into that frame of mind allows you to tell yourself “It’s alright to make mistakes “, and in doing so, you’re not as fearful to venture into unknown territory.

Assignment structures were well planned and thought out. I found that each assignment contained what we needed to cover and always had an element of exploration which was fantastic. Tests including any midterms or finals were fair. After the debugging session of the course, I felt that it really should have been introduced sooner. This also applies to the Java course. Immediately after that section, I was fully utilizing the debugger when I found myself spending more than 5 minutes searching for an issue. Nine times out of ten, I always found the issue with its assistance. Scripts and web applications were certainly more complicated as the course carried on, which made the debugger a more useful tool, but I can still see its use in teaching script / application execution flow a vital point.

I mentioned collaboration in the introduction of this paper and I wanted to elaborate on that a little more. As we know when we develop an application, there is not a single defined way to complete a task. There are recommended ways because of best practice standards and security related rationale. If we took a single assignment completed by an entire class (so each member of the class completed it separately) and compared each member’s application structures, how many different methods do you think might be used? I found that we never really discussed each other’s development choices openly. It’s as if our choices and approaches to a problem were a secret to be locked up in a vault forever, so no one would ever know. I hope in the future that as students in the software development world, we can share each other’s experience and knowledge more openly. I believe that there is great benefit for all in this. It could be compared to how we pull specific pieces of data from a data source and piece them together to build useful information.

If I’ve gained anything from the hours put into assignments, researching on the web, and struggling with logic errors, it’s that software development is no easy task. It changes constantly, so you are always in learning mode. It’s a skill comprised of a number of skills that must be flexed and put to use regularly. If you do not do this, then your back at square one (much like not going to the gym for a long period of time) relearning what you have forgotten. Yet, as new recruits in the IT industry, we are not alone. There are experienced software gurus that endure the same hardships we endure. We are all human after all.