CS 499 Self-Assessment

Adam Kulp

Southern New Hampshire University

When I began the journey to pursue a degree in computer science; it wasn’t simply an on-the-fly decision, nor was it one that I was unsure about nor was it a decision that I would come to regret. From a very young age, I have always loved technology and everything it had to offer; the advancements that have been made in just a few short years are nothing short of remarkable and I would like to play a role in future technological advancement. When I began this journey however, I wouldn’t know exactly what to expect; I only knew that it was something that I truly wanted to do. Of course, basic education classes such as Math, Science and English would occur first which at the time I found to be a bit mundane but looking back each of these courses would genuinely help with most if not all future computer science courses. Handfuls of different languages, philosophies and systems would be introduced that all required those “mundane” courses to truly master. I would learn to code in C++, Java and Python through the creation of many different projects (both for school credit as well as on my own for self-progression); I would delve into the world of database management with MongoDB and MySQL and ultimately come away with a relatively deep understanding of how databases work and I would even learn to create 3D objects in space through the use of OpenGL; definitely with the help of all of those previous math classes. These concepts are just the tip of the iceberg when it comes to concepts that I have learned. Others include cyber/web security, different operating environments, software testing as well as emerging architecture and technologies to name a few. With that being said, I felt that it was of utmost importance to provide working examples or artifacts that would fully represent the robust knowledge that has been obtained through these courses.

The first artifact was created when I was approximately halfway through the program as a requirement when applying for my first junior developer position. This would be the very first standalone and fully functional program that I had written in Java; I attempted to pull out all of the stops by making it as complex and intuitive as possible. This particular program was intended to calculate taxes from predetermined items and print a receipt (just like a cash register). The original directions only required a specific set of items to be calculated but I took it a step further as to either allow user input or read in a file to allow for greater versatility of this project. It would allow for calculations of sales tax as well as duty tax (if applicable) and print a receipt. This particular project was chosen not only because it was accomplished in a one-week time frame which I find to be remarkable for not knowing java at all but also because of the many concepts that were implemented in the code. OOP concepts such as data abstraction, encapsulation, inheritance and polymorphism were used, many basic java constructs were utilized, exception handling and string handling was used and a myriad of classes, methods, decorator patterns and interfaces were used which I feel showcases a relatively good understanding of the java language. Looking back at this project, while functional, it definitely needed some work to get it to where it is now. I had to go back through the entirety of the code and properly format and comment, work out a small bug which was found through Junit testing that wouldn’t always automatically assess duty tax when an item said “import” or “imported” as well as create a README file to ensure that anyone utilizing the program would understand how it works. Overall, this program turned out quite well and I feel shows a well-polished java program that implements good software design and engineering as well as properly used and implemented algorithms and data structure.

The second artifact was created approximately one year ago when I was playing around with python and PYgame. At this point I had very little experience with python and simply created this snake type game in an attempt to learn a few python structural fundamentals in a fun way. The result was a working snake app that would allow the user to control a snake to eat the apple (with standard W,A,S,D controls) and grow. The game would be over and display a message when out of bounds was reached. The code could also be modified to adjust difficulty level. I felt that overall, this particular project was executed quite well both visually and systematically speaking. It was a project that I felt good about and one that I feel could definitely be functional as is but also leaves a lot of room for growth and development in the future. This particular project taught me quite a bit about the python language including but not limited to the creation of a window size and initializing it, creation of a frames per second controller, assigning color, checking for errors, assessing game variables, keeping a score tally and creating objects that can be consumed upon collision such as the apple with the snake. Just like the last artifact documentation and commenting was my downfall so I had to go back and organize the code, create proper comments and provide a README for anyone that may be using this application. While I wasn’t able to create a game menu and add additional games to this menu prior to the final project being due; I was able to add an options menu that would allow users to change difficulty setting without the need to edit the code itself. I will continue to work on this piece to learn additional concepts in the python language. I believe overall though, this piece showed many positive design and structural elements and is a properly functioning standalone game.

The third artifact was created approximately six months ago when I took a class on database and database management. This particular project utilizes python along with mongo DB in an effort to display filtered animal information based on user’s needs. In this particular instance the program was being created for a company called Grazioso Salvare which finds and adopts animals to train for various purposes. This particular project contained data for a myriad of different animal types but dogs were the focus for the project. This data could be sorted based on breed, color, type etc but buttons were also built in that would auto sort for specific animals that were considered to be premier rescue animals as per the project guidelines. This project would also utilize the data within the database to display a map and pie chart for the data. The main purpose of the project was to create a program for Grazioso Salvare that could utilize shelter data in an effort to easily look for and obtain shelter animals that would be great at rescue training. While I without a doubt struggled on this project, I felt it was one that could showcase my ability to comprehend a multitude of concepts such as python, mongo DB, the use of databases and dash components, integration of charted data, basic CRUD operation and implementation as well as security features associated with mongo DB and database management. There were a few problems with this program as well which mostly stemmed again from organization. A README file was created and all code was gone over and commented and properly formatted. A small problem was also occurring with the radio buttons that filtered animals but has since been fixed; resulting in what I believe to be another quality project that shows many concepts such as those stated above as well as actual use and manipulation of database data.

In all, while all three of these projects could always use upgrades, changes and advancements; I find them to be fundamentally sound and are a snippet of many of the concepts learned throughout the curriculum. The good thing about computer science and one of the many reasons that I chose this path in life is for the very fact that there is always room for improvement, growth and learning. The biggest thing that I learned through this project was to follow good coding practice and utilize testing for all programs prior to release. Having to go back and comment, format and test is a very time intensive and tedious task that would be much easier going along. While I polished these artifacts for the sake of the project, I will continue to work on them and many more as well to further build upon my skills for the future. It’s quite incredible to me that I went into this program knowing very little about computer science and coming out with proficiency in 4 languages as well as an understanding about databases and security among other things. While there is still much to learn; I feel that this curriculum has setup a foundation of understanding and a sense of confidence in myself that may allow me to gain employment within the computer science field at some point in the near future.