Jordan Colbert

CS 499

June 19, 2022

Southern New Hampshire University

Self-Assessment and Reflection

Working throughout the Computer Science program at SNHU has benefited me greatly in many ways. I had no real experience in software development previously, nor did I truly have a grasp on the basics of IT and how things worked in the tech world. Various courses taught me skills in the world of computer science, such as the course that introduced C++, and the one that taught me about full-stack development in Python. A key source I knowledge I gained was the software development lifecycle, or SDLC. I learned the process in which software is developed, from conceptualization all the way to deployment and maintenance. Scrum and agile methods came into play as well—this taught me about collaboration with others through frequent meetings and sharing of ideas and developments in a regimented cycle during the SDLC, along with development of user stories and gathering of information to go between developers and stakeholders.

Of course, as the artifacts suggest, software design, algorithms, data structures and databases, along with security, came into play as well. Through the learning of various languages, each category came into play. I mostly learned design through my early Java classes, as this taught me about methods and object-oriented programming. An early project involving methods to create a list of movies that relied on a loop with a “menu” was my first experience with being concerned with design and why programs are developed modularly. Algorithms and data structures came through both Java experience and C and C++ courses. I found myself having to consider how I structured my programs and what the logic driving them does. Efficiency and consideration of the goal became paramount. This came into play when working on embedded programming in a project using a microcontroller programmed to function like a thermostat. I had to be concerned with how I handled my coding for concerns about storage overhead and simplicity due to the low specs of the device. Finally, databases came into play with MongoDB, which was frequently paired with another language, such as Python. Manipulation of databases was an entirely different skillset, with commands not resembling anything I had seen prior. I had to adapt and learn how to change data that was not immediately visible and think in terms of tables rather than methods. An example was a project where I had to pull data from preexisting databases and move it into a live application in Python and figure out how to format it for proper display. Finally, lessons in security came in a course delving into defensive programming in Java, where I learned to prevent overflow or outside manipulation with assertions or checks on variables. All these principles came together when developing this portfolio and the artifacts within.

Each artifact exemplifies a specific skillset and category. As stated, these categories all come together to make up the knowledge I gathered in the course. The three categories represented are software design, algorithms and data structure, and databases. Through each project, I had to learn much about each concept. There is some overlap in the examples—my software design artifact could easily fit into the data structure category, and vice versa. This only proves how impactful these projects were in my progress in this field. Taken as a whole, each artifact is vital to my development and reflects my growth, not necessarily in order. I believe my examples that follow will demonstrate exactly what I have accomplished at SNHU.