**Professional Self-Assessment**

Ian Pruter

Southern New Hampshire University

CS 499: Computer Science Capstone

Professor Brooke Goggin

August 14, 2021

**What I’ve Gained**

Throughout the SNHU Computer Science Program I’ve been introduced to a range of skills and concepts necessary to be successful in the field. Coding best practices was a focus early on to ensure my work adhered to standards that would make it easy for others to maintain or incorporate into other programs. Using concepts like organizing code into modules, classes, and methods made my code more robust and efficient by reducing redundancy and hiding data within the classes. Another focus was the development cycle where I gained a holistic understanding of the process and learned how to set realistic expectations for stakeholders to help them make informed decisions and prioritize program features to deliver the most optimal program that meats deadlines and customer requirements. The realistic expectations also apply to the project manager and development team to set feasible deadlines in the production schedules. I learned about iterative development cycles where teams assess progress and obstacles, add or edit features, and test all changes. I also learned about industry standard design techniques such as UML diagrams including class diagrams and activity diagrams. Pseudocode is another design technique I use regularly to plan the programs logic without getting caught up in syntax related issues. I learned how systems of hardware and operating systems allocate resources and organize processes so I can code in a way that best works on the systems it’s intended to be used on.

Discrete math taught me computer concepts like how to optimize programs and structure data. I learned how to collaborate with classmates using version control tools like Bitbucket and EGit to develop a jukebox application together. In the same class, I learned about using abstraction to hide complexity in a class in anticipation for future changes to it or its objects. We also learned to code review individuals within our designated team and create productive experiences by applying collaborative best practices such as keeping the critiques constructive and acknowledging strengths in the student’s work. Another class thoroughly focused on the value of testing at various levels like documents, units, control structures, black box, and catching bugs before they become exponentially more expensive. A class on algorithms and data structures thoroughly explored the pros and cons of the different structures. A class on clients and servers went over different protocols and how to mitigate various security breaches. This helped me adopt a security mind set where I consider the consequences of neglecting constraints and exceptions in my design and code. Lastly, I learned how to render images in the GPU using vertices indices and shader files in OpenGL.

The course gave me a depth of knowledge that sets me apart with the skills to

* Implement strategies for diverse audiences to collaborate and foster effective, organizational decisions in the field.
* Communicate effectively through various mediums most appropriate for specific audiences and contexts.
* Effectively solve problems with appropriate computing solutions, algorithmic principles, and best practices for specific contexts.
* Accomplish industry-specific goals using well-founded innovative techniques and industry standard tools
* Approach design, development, and testing with a security mindset to mitigate design flaws and ensure privacy and enhanced security of data and resources.

**ePortfolio**

I applied a lot of these skills again while putting together the ePortfolio in the capstone project. The projects had me focus on engineering and enhancing code, data structures and algorithms, databases, and security. I made a video reviewing three projects for errors, adherence to best practices, and security vulnerabilities. The three projects were then fixed and enhanced in various ways to demonstrate my ability to understand and implement some of the most important concepts in the computer science program. I re-engineered a project written in Java to Python where the modules can be incorporated into python projects easier. During the design phase I had also fixed an architectural issue where the list was a global variable. I fixed this by making the list its own class containing the CRUD functions that edit it. This removed the security and organizational issues that come with global variables while organizing classes in a way that make them easier to edit and re-use.

In the second enhancement I changed the linked list to a binary tree and added a balancing algorithm so the list searches in a logarithmic amount of time relative to the size of the list as opposed to a one-to-one relationship of time and list length. During the planning phase, I applied best practices in evaluating computer solutions by analyzing the pros and cons for different data structures. A tree and balancing algorithm are needlessly complex for small data sets but much faster at searching through data sets as their size increases. The enhancement called for a more efficient data structure accompanied by an algorithm, so the binary search tree was the perfect solution. For the third enhancement I added a database and an NLP feature that allows the user to search for contacts in the database by segments of the contacts first or last name. I reduced security vulnerabilities and ensured data was validated by password protecting the database and containing code that communicated with the database in try and catch statements. This reduced users’ ability to find vulnerabilities in exceptions to expected program use. These enhancement projects were accompanied by narratives further demonstrating what I learned during the process. Together, these projects demonstrated my aptitude in developing quality programs from critical concepts accumulated throughout the computer science program.