**Professional Self-Assessment**

Student Name: THOMAS SEIBERT

Southern New Hampshire University

CS 499: Computer Science Capstone

Link to ePortfolio: <https://trs1225.github.io/index>

10/10/2025

**Self-Assessment: Building knowledge, skills, and a career pathway in Computer Science**

**A computer and phone on a table

AI-generated content may be incorrect.**

Welcome to my ePortfolio. Here, I will use examples to demonstrate how the advancement of my software engineering skills, as well as the knowledge that I have gained in the process of completing the Computer Science Bachelor’s Degree program at SNHU have been fundamental to progressing on the pathway of an engineer. This coursework has allowed me to learn in-depth of various programming languages such as Python, Java, and C++, as well as how to build data structures and implement algorithms, along with other software development projects including full stack applications, 3D graphics design and rendering of scenes, as well as Android mobile application architecture. Aside from these technical skills, I also learned how to operate within the Scrum framework, an agile methodology employed by development teams to manage the Software Development Lifecycle, in addition to various other methodologies pertaining to the project life cycle.

At this time, I will provide examples of completed coursework from my degree program with screenshots to demonstrate various skills and knowledge which were acquired in the process of completing my degree program.

**Full Stack Development**

In a course titled "Full Stack Development", I built a full stack web application utilizing MEAN (MongoDB, Express, Angular, Node.js) components. In this case, the web application that I developed allowed users to select and book various travel vacation packages for a client. During the beginning stages of application development, all the application logic had been handled by the server, with webpages rendered server-side, and thus presented to the browser (client-side) for display. Controllers, models, routes were developed in JavaScript to build the user interface of the application. Express was then used to retrieve and render static HTML via the utilization of Handlebars (HBS) template views, which were populated using data from the database application, MongoDB.

We later transitioned this application into a single-page application (SPA), which utilized API calls to fetch and manipulate data. To accomplish this, we used Angular as the basis for this coding framework. SPA is a form of web application development where all the application rendering happens on the client-side. This minimizes client-server traffic. After the transition to SPA, the only calls which were necessary to make to the server-side were made to retrieve data stored in the database.

Here is a screenshot of a sequence diagram made for this web application referring to the path by which a user's actions (or route) taken within their browser is handled by the application itself. The route, in this case, refers to the web address and link clicks that a user can perform within the app. These actions included signing in, viewing website information, and administrative actions:

A diagram of a diagram

AI-generated content may be incorrect.

*(Screenshot of my full stack web application’s sequence diagram from the course titled “Full Stack Development”).*

* Link to Github Repository containing this full stack web application: <https://github.com/TRS1225/CS465-Full-Stack-Development/>

**Computational Graphics and Visualization**

In a course titled "Computational Graphics and Visualization", I built a 3D scene, constructed from a real-life photograph, which featured basic household objects of various shapes. This course was my first introduction to 3D graphics building and rendering. In the scene that I built, there is a roll of tape, an audio interface, a cup, a hacky sack, and a picture frame all placed on a desk. I developed this project using C++ in Visual Studio utilizing OpenGL. Here is a screenshot of my constructed 3D scene to demonstrate concepts that I learned during this course:

A computer screen shot of a computer screen

AI-generated content may be incorrect.

*(Screenshot of my 3D scene from the final project of the course titled "**Computational Graphics and Visualization”).*

* Link to Github Repository containing this 3D graphics project: <https://github.com/TRS1225/CS-330-Computational-Graphics-and-Visualization>

**Mobile Architecture and Programming**

In a course titled "Mobile Architecture and Programming", I developed a mobile Android application designed to track, manage and monitor inventory stock within a client’s warehouse. This mobile application, written in Java, was designed to be as user-friendly as possible, accessible and usable by anyone with an Android smartphone. Downloadable from the Android app store, it allows users to create an account, login, add inventory items to a database, delete inventory from the same database, and receive SMS notifications which indicate that their warehouse inventory is low on any specific item. It also allows for simple navigation between screens using buttons, and it also features a text-based indicator which displays the user's SMS permission status, and whether the user has chosen to receive SMS notifications. Here are screenshots of the Inventory app’s icon, as well as a code snippet from the program itself:

A clipboard with check marks and a box

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

*(Screenshots of my Inventory of Warehouse mobile app icon, as well as a code snippet from the program itself, from the course titled “Mobile Architecture and Programming”).*

* Link to Github Repository containing this Android mobile application project: <https://github.com/TRS1225/CS-360-Mobile-Architecture-and-Programming>

**Categorical Knowledge and Skills**

Now that I have demonstrated various software development skills acquired in the process of completing coursework within the degree program, I will highlight relevant topics broadly within the field of Computer Science and how my skills align with these topics:

* **Collaborating in a team environment:** In the course titled, “Software Development Lifecycle”, I learned how to manage the software development life cycle within the Scrum framework, and how teams work together using this agile methodology through the utilization of the daily Scrum, sprints, and frequent testing cycles, to achieve specific goals.
* **Communicating with stakeholders:** Within the context of agile methodology, I learned when and how to communicate with project stakeholders. This is to be done at key points within the software development life cycle. These key points include sprint planning, where stakeholders have the option to provide input on priorities to the Scrum team, determining which items from the product backlog will be worked on in the upcoming sprint. Sprint reviews, where stakeholders are invited to inspect an increment of work completed to provide various feedback. Presentations, at minimum, should also be made to stakeholders at the start of a project, at key project milestone completions, and at the final deliverable phase (the end of a project).
* **Data structures and algorithms:** My proficiency in this area has been made evident in projects where maximized efficiency of software solutions was paramount. Over the course of my degree program, I developed and then later enhanced algorithms for loading and sorting data into designated structures within applications, the purpose of which was to improve performance and reduce unnecessary complexity.
* **Software engineering:** Through designing and developing various applications over the course of my degree program, I became skilled at building efficient solutions using multiple programming languages such as Java, Python, and C++. In so doing, I also learned to implement non-coding strategies such as the construction of pseudocode to develop application blueprints, for the purpose of implementing complex logic to achieve application goals. These skills, along with a focus on designing efficient application-specific methods, have prepared me to handle real-world engineering projects.
* **Database:** Over the course of my degree program, I learned how to communicate with a database server effectively and decisively. I worked with both MySQL and MongoDB, a NoSQL database. In so doing, I was able to communicate and interact with databases by executing various SQL commands, including but not limited to the four primary statements, as well as other data aggregation techniques. I also used the Postman application in conjunction with MongoDB Compass to GET, POST, PUT, and DELETE data within the context of full stack web applications.
* **Security:** In terms of implementing security concepts through code in compliance with security testing protocols, I learned how and why to apply encryption technologies and techniques to communications. Specifically, in a course titled “Software Security”, I refactored a program’s code to implement an encryption cipher (SHA-384) in the form of a checksum validation to protect a client’s sensitive financial data. In addition to this, I implemented HTTPS to prevent the potential for exploitation of vulnerabilities while data was in transit, as well as an OWASP Dependency Check to further reduce the potentiality of plug-in exploitation. Altogether, these changes amounted to a more secure program for the client, and a demonstration of my skills and abilities as far as software security is concerned.



**Artifact Enhancements**

Now, I will summarize how a select few artifacts, which have been enhanced for the purpose of demonstrating a full range of my Computer Science talents and abilities, fit together and inform this ePortfolio. Each artifact stands alone and contributes to a narrative of my skills and abilities within the field.

The combination of these artifacts consists of three projects fitting within the criteria of three different categories, namely that of **software engineering and design**, **algorithms and data structures**, and **databases**.

* Link to Github Repository containing all artifact enhancements: <https://github.com/TRS1225/TRS1225.github.io>

**Software Engineering and Design**

For example, the GrandStrandSystems project, which was the final project from a course titled “Software Testing, Automation, and Quality Assurance”, is a mobile application enhanced to demonstrate satisfactory proficiency of skills within the category of **software engineering and design**. This program utilizes a unit test framework to verify that the individual components of the program function as intended. The primary enhancement of this artifact is that it was originally written in Java, and was then converted entirely into Python, while additionally more descriptive comments and component names were added as well. By implementing these enhancements, I have aligned with multiple outcomes which were expected in my degree program, including:

* Employment of strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science.
* Demonstration of an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.
* Development of a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

Here is a screenshot of the “GrandStrandSystems” program running after artifact enhancements were implemented:

A screenshot of a computer

AI-generated content may be incorrect.

*(Screenshot of the “GrandStrandSystems” program running after artifact enhancements have been implemented, displaying that all unit tests have passed in PyCharm, for the category of Software Design and Engineering).*

**Algorithms and Data Structures**

The next artifact enhancement, relevant to the category of **Algorithms and Data Structures**, was the final project from a course titled "Data Structres and Algorithms". It is a program written in C++, designed to be a course planner made for Computer Science department advisors. This program organizes and prints various course information including course numbers, titles and their prerequisites. It utilizes a structure object to contain course information, and an unordered map as a hash table to store a course's number as its key. It allows users various menu options, including the ability to load course information from either a hardcoded array or a file, print a list of courses, and display the details of a course, including its prerequisites.

This artifact enhancement demonstrates an ability to understand and improve not only the efficiency and time complexity of a specific algorithm, but also the program as a whole and how said algorithm works with the rest of the program. The primary enhancement consisted of a restructuring of the “loadDataFromFile” method to pull course information (number, title, prerequisites) from lines in a file and put them into a structure object using substring operations, instead of a vector. Consequently, the algorithm pulled course prequisite information in a single pass instead of multiple passes, which reduces RAM usage. In addition to this primary enhancement, comments, function names and variables were made more descriptive where possible.

With the completion of this enhancement, I aligned with the following expected outcome, namely:

* Design and evaluate computing solutions that solve a problem using algorithmic principles and computer science practices appropriate to its solution, while managing the trade-offs involved in design choices.

Here is a screenshot of “Project Two” program running after the artifact enhancements have been implemented:

A black screen with white text

AI-generated content may be incorrect.

*(Screenshot of the program “Project Two” running, after artifact enhancements have been implemented for the category of Data Structures and Algorithms, displaying the functionality of the “loadDataFromFile” method).*

**Databases**

For the category of **Databases**, and the final artifact enhancement, I chose to enhance a program titled "Binary Search Tree" from the "Data Structures and Algorithms" course. This is a program written in C++, designed to perform various functions related to data from a CSV file containing information on a bid auction, which functions as its database. These functions within the program include the ability to load the data, print all bids, find a bid, add a bid, remove a bid, as well as the primary enhancement which is to incorporate data mining via the utilization of a Binary Search Tree class. For the purposes of this enhancement, I have added to, modified, and thus improved every aspect of the program.

The primary enhancement demonstrates an ability to incorporate data mining in a way that is straightforward and easy to understand. This was done through the calculation of statistics associated with the program's associated database, in this case, a CSV file. Also, I have improved the program all around by adding more functionality and validating user input, as well as incorporating more descriptive comments throughout the program.

Specifically, the incorporation of data mining allows this program to iterate through bid dollar amounts, and calculate various results, such as what the average bid is, highest and lowest bids, sum of all bids, and the total number of bids. From the main menu, users can choose to incorporate data mining of these statistics by choosing a numeric option.

By implementing this enhancement, I have aligned with the following expected outcome:

* Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.

Here is a screenshot of the “Binary Search Tree” program running after these enhancements have been implemented:

A computer screen with white text

AI-generated content may be incorrect.

*(Screenshot of the “Binary Search Tree” program running after artifact enhancements have been implemented for the category of Databases, displaying the functionality of incorporating data mining.)*

**Summary**

In summary, these artifact enhancements and previous coursework highlights reflect my dedication to not only improving my own technical skills, but also to increase my own competency and understanding of how collaborative, teamwork focused dynamics work within the field of Computer Science.

My aspiration in this ePortfolio is to demonstrate that I am not only educated in how to build and enhance software applications, but also that I am equipped with a practical focus on efficiency, to align with a path of success in my future career. This ePortfolio is more than a highlight of my previous coursework, it is a declaration of my commitment to contribute to the field of Computer Science.

A wooden blocks with letters on it

AI-generated content may be incorrect.