This artifact was a project from my CS 320 course. It is a Java contact list with unit tests. The main files, Contact and Contact Service, create the contact object, store it in an ArrayList, and manipulate it using CRUD principles. The test classes use static data and unit tests to test the functionality of these main classes.

            I included this in my portfolio because it is a clear showcase of my ability to understand and effectively utilize a common data structure. It also has the added benefit of highlighting my quality assurance and security skills through the unit tests. I improved the artifact by porting it into a different language and using a vector instead of an ArrayList. Choosing a vector also indicates good security protocol because these are considered safer than raw arrays and are a good preventative measure against buffer overflow vulnerabilities. Although the Java ArrayList and C++ vector function similarly and are seen as equal, showing that I can successfully understand and use both in a program is a good addition to my portfolio. I also included a second version of the project enhancement where I chose to create a contact list with a menu. This version exhibits more of my security and quality assurance skills because I had to include multiple forms of input validation to ensure only safe data is processed and stored.

            With these enhancements, I did meet my intended course outcomes and think I’m on track to meet all five through my entire portfolio. Therefore, as of right now, I don’t plan to change or update my outcome-coverage plans.

            The overall process of enhancing the artifact wasn’t as difficult as my other project enhancements because I’m more comfortable and confident with the C++ language, but I still learned a lot and had to work through some issues. It’s been a while since my data structures and algorithms course, so this project prompted me to revisit the basic concepts and the importance of time complexity in programming. I learned how to utilize data structures best, how to choose the appropriate one for my program, and when I would use others. As for the issues I faced, the first problem was that I had never used the Google Test framework or written Google Tests, so setting it up and learning the syntax was a small hurdle. Most of the test framework was similar to the JUnit tests, so that was easy to overcome, but I also had issues with the software and errors in the header files. I utilized Visual Studio and Microsoft forums, along with the Microsoft Learn website to see if anyone else had any issues; I realized my Visual Studio and Google Test versions weren’t compatible, but this was an easy fix. Fortunately, I didn’t face any more significant issues after this.