**CS 499 – Milestone Three Narrative**

**Briefly describe the artifact. What is it? When was it created?**

The Enhanced Numeric Overflow Detection Tool is a C++ program I made for CS 405: Secure Coding back in 2023. Originally, it was basic - just checked if adding two integers would cause overflow by comparing against INT\_MAX and INT\_MIN. It only worked with regular integers and didn't really show any advanced programming skills.

**Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?**

I chose this project because it gave me a chance to take something simple and make it way more complex and useful. The biggest change was turning those basic overflow checks into template functions that work with different number types like int, float, double, and long. This was tricky because I had to learn about templates, which I'd barely used before.

I also added a vector to store test results using a custom struct I created called TestResult. This shows how I know how to pick the right data structure for organizing information. Instead of hardcoded values, I used std::numeric\_limits which makes the code work better across different systems. I broke everything up into organized sections with namespaces too.

The coolest part was adding a testing system that automatically checks if everything works correctly with different number types. It really shows that I understand how to validate algorithms properly.

**Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

Yeah, it met what I planned. The template stuff shows I can design solutions using good algorithmic principles, which covers Course Outcome 3. Using modern C++ techniques like templates demonstrates the innovative tools from Course Outcome 4. The whole point is still preventing security problems from overflow bugs, so that hits Course Outcome 5.

I don't need to change my plans because this turned out better than I expected. I learned way more about templates and algorithms than I thought I would.

**Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

This project was a real learning experience. Templates were completely new to me and figuring out how to make one function work with totally different data types was confusing at first. I had to understand how the compiler creates separate functions for each type you use, which is cool once you get it.

Learning about data structures was interesting too. I found out why vectors are better than arrays for this kind of thing and how to make custom structs that work well with the standard library stuff.

The hardest part was dealing with signed versus unsigned numbers. They behave totally differently when they overflow - unsigned numbers just wrap around to zero, but signed overflow is undefined behavior in C++, which basically means anything could happen. I had to write different logic for each type and use std::is\_unsigned to figure out which algorithm to use.

Another challenge was making sure all the pieces worked together without slowing things down. I had to think about how the logging would affect performance and make sure my data structures weren't creating bottlenecks.