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

**Category One**: Software Engineering/Design

Artifact Name: Secure Coding

Origin: Coursework in CS 405 - Computer Science program

Category: Software Engineering/Design

The artifact is a C++ file that contains NumericOverflows.cpp.

The "NumericOverflows.cpp" file is a C++ program that shows methods for detecting and preventing numeric overflows and underflows. Template functions are for adding and subtracting numbers to check for overflow and underflow. The program tests signed, unsigned, and real numbers. It presents overflow and underflow findings to the user. The code handles arithmetic operations, data types, and numeric limit error detection.

**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 software development? How was the artifact improved?**

The selection of the artifact and the inclusion of it in my ePortfolio is to show proficiency in various programming languages. Specifically, it shows my skills in translating a C++ code to Python code while preserving the functionality and logic.

The following components of the artifact shows my skills in software development:

Numeric Operations: The code includes arithmetic operations such as subtraction and addition. By converting this code to Python, the skills demonstrated include numerical calculation and management of overflows and underflows of integers.

Flow Control: The code contains for loops and if statements to control the execution flow. The translation of this code to Python demonstrates knowledge of flow control structures and iteration management.

Type Handling: The code performs type-specific operations on various data types, such as integers and floating-point numbers. The Python translation demonstrates proficiency with a variety of data types and with type verification.

I improved the artifact by enhancing readability, reducing syntax complexity, enabling platform independence, easing debugging, and integrating into a larger community and ecosystem. These enhancements increases the efficacy, maintainability, and adaptability of the artifact in a Python-centric development environment.

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

The course objective is to showcase an alignment of the artifact with software design/engineering. This objective was met by conforming to the software design/engineering principles of modularity, encapsulation, error handling, testing, and documentation. These are consistent with the overarching objective of devising robust, maintainable, and reusable software solutions that adhere to software engineering best practices.

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

During the process of enhancing the artifact, I was able to improve my proficiency in both languages. Translating the C++ code to Python requires a solid understanding of both languages. The procedure improves my proficiency in C++ and Python, including knowledge of their syntax, data types, control structures, and library utilization. It facilitates familiarity with language-specific characteristics and procedures.

**Challenges:**

The challenge I had was adapting the C++ code to another language while dealing with various limitations between the languages. Different programming languages have their unique features, libraries and limitations. Bearing this in mind during the code conversion was a challenge for me.