### **CS-410 – Algorithms & Data Structures Narrative**

#### **1. Artifact Overview**

**Title:** Binary to C++ Conversion and Analysis  
 **Original Course:** CS-410 Reverse Software Engineering  
 **Created:** Spring 2025

This artifact was originally a binary analysis project where I reverse-engineered a compiled .o file into assembly, analyzed its functionality, and rewrote it in secure, functional C++ code. The program handled conditional logic, memory access, and control flow to determine output based on user input.

#### **2. Justification for Inclusion**

I selected this artifact to represent my skills in algorithms and data structures because it demonstrates my ability to analyze low-level logic and translate it into optimized, high-level C++ code.

Enhancements I made:

* Improved efficiency of conditional branches using simplified decision trees.
* Replaced nested if-else blocks with switch statements for better clarity and runtime behavior.
* Added input validation and boundary checks to ensure security and proper logic flow.
* Restructured loops for better control and reduced complexity.

These changes improved the algorithm's clarity, performance, and reliability, aligning with course outcomes around solving problems using efficient algorithmic principles and managing trade-offs in implementation.

#### **3. Enhancement Reflection**

While enhancing the original artifact, I learned how to optimize logic flow for readability and runtime efficiency. I also became more comfortable recognizing patterns in binary and assembly that map to higher-level constructs.

A major challenge was preserving the original binary behavior while improving the C++ structure. I used tracing and test inputs to confirm the enhanced logic produced identical results with better performance and maintainability.

This artifact now showcases my ability to work at both low and high levels of abstraction, design clear algorithms, and implement safe, optimized logic for real-world problems.