**Portfolio Reflection**

Throughout this course, I have gained a much deeper appreciation for the importance of adopting a secure coding standard early in the development process. Following a standard like the SEI CERT C++ Coding Standard ensures that security is built into the foundation of the application rather than treated as an afterthought. Waiting until the end to address security often leads to costly rework and missed vulnerabilities. Concepts like input validation, proper memory management, and avoiding deprecated functions highlight how critical it is to write secure code from the start.

Evaluating and assessing risk also became a major part of my learning. I now understand that risk assessments are necessary to prioritize vulnerabilities and determine whether the cost of mitigation is worth the investment. Using severity, likelihood, and remediation cost to assess each threat allows developers to make informed decisions about how to allocate resources most effectively. Risk assessments also tie directly into our security policies and help justify why certain controls and measures are necessary.

Zero trust was another important concept that reshaped how I think about security. The "no one is safe" mindset means that trust is never assumed, even for internal users. Every access request must be verified, and minimal privilege should be the norm. Implementing zero trust increases security overall but also requires developers to ensure constant authentication, authorization, and accounting processes are in place throughout an application.

Finally, implementing and recommending security policies, like encryption policies and Triple-A (Authentication, Authorization, and Accounting), are essential for a secure development lifecycle. By establishing clear standards for encryption at rest, in flight, and in use, as well as guidelines for user management and access control, organizations can enforce consistent protection across systems. Writing and maintaining these policies ensures that all development teams have a shared understanding of how to integrate security at every stage. This course has reinforced that security is not a feature to add later — it must be part of every decision we make during development.