Kenyhacta Busby

CS 499

11/22/2024

Milestone Three

1. **Briefly describe the artifact. What is it? When was it created?**  The artifact is a security-focused project created in C++ as part of the CS 405 course. It involves creating uniform implementation guidelines, presenting a security policy guide for Green Pace, and ensuring that the development team adheres to best practices. It includes writing unit tests to identify vulnerabilities and addressing risks using 10 guiding security principles.
2. **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?** The ePortfolio was chosen to include this artifact because it shows how well I understand and use algorithms and data structures. It shows how I solved problems like allocating memory efficiently and handling input safely. Both are very important in modern software development. The validation logic, performance logging, and retry mechanism show that I can use data structures and algorithms to make programs more reliable and efficient. The retry mechanism shows how control flow and algorithmic problem-solving can be used to limit attempts at invalid input, making sure that errors are handled correctly and making the user experience better. Performance logging shows that I can use time-based metrics to analyze and rate how efficiently memory is allocated, giving me information about how well the program works. Input validation and SQL injection prevention also show how well I can use string manipulation and logical checks to make sure that programs run safely and reliably, keeping the system safe from malicious inputs while keeping it functional. Together, these improvements show that I can use effective algorithms and data structures in real-world situations. Retry limits were added, memory allocation logic was made better, and performance logging was added to make the artifact more efficient. Implementing a structured retry counter, checking user input for SQL injection risks, and adding time tracking to measure allocation performance were all needed for these improvements. These changes turned the artifact from a simple code into a complete program that shows how algorithms and data structures can be used in real life.
3. **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?**  No, the artifact did not initially meet the course outcomes planned in Module One because the algorithms and data structures focus was too similar to the security enhancement in the first category. To address this, I updated the artifact to better align with the second category by ensuring a stronger focus on distinct algorithmic principles and data structure implementations. These updates emphasize input validation, memory allocation, and task management in a way that highlights structured problem-solving. By leveraging Python’s built-in data structures, such as lists, and integrating performance logging with the time module, the artifact demonstrates the ability to use tools effectively and creatively. These changes ensured that the artifact now aligns closely with the targeted outcomes and clearly differentiates itself from the first category by focusing on algorithms and data structures.
4. **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?** I realized how important it is to thoroughly document your code after modifying and enhancing the artifact. Documentation that is clear and to the point not only makes the code easier to read. It also makes it much easier to find and understand later, especially when making changes or fixing bugs. One challenge I faced was making sure that the updates made it clear which ones were focused on algorithms and data structures and which ones were focused on security. To make sure this fit with the second category outcomes, it needed to be carefully planned, and the artifact had to be reevaluated. Putting the improvements to the test to make sure they work while staying simple was another problem. Even with these problems, the process made me realize how important it is to code in an organized way and solve problems in a structured way.