CS-499 Milestone Three

Darrell Lindsey

09/27/2025

Milestone Three represents a pivotal opportunity to demonstrate my evolving expertise in algorithms and data structures, which are foundational to the discipline of computer science. For this milestone, I selected a custom implementation of a binary search tree (BST) that I originally developed during prior coursework and later enhanced as part of my final project in CS 499. This artifact is a core component of my Android inventory application, where it serves as the underlying data structure for organizing, sorting, and retrieving inventory items based on multiple criteria such as name, expiration date, and quantity. The BST implementation includes the InventoryBST and BSTNode classes, which together enable efficient recursive traversal, dynamic sorting via comparator injection, and integration with the app’s user interface. The decision to include this artifact in my ePortfolio was driven by its complexity, relevance, and the breadth of skills it showcases. It reflects not only my ability to design and implement algorithmic solutions but also my capacity to refactor and optimize code for performance, maintainability, and testability.

The enhancements I performed for Milestone Three were guided by the code review and planning process completed earlier in the course. These enhancements included refactoring the BST to support multiple sort modes using a flexible comparator strategy, improving traversal efficiency by eliminating redundant recursive calls, and expanding unit test coverage to validate edge cases such as duplicate entries and null-safe comparisons. I also ensured that the BST logic was compatible with Kotlin Coroutines and Android lifecycle constraints, which required careful consideration of threading and asynchronous execution. These improvements align directly with the Computer Science program outcome focused on designing and evaluating computing solutions using algorithmic principles and appropriate standards. Furthermore, the integration of the BST into a full-stack mobile application demonstrates my ability to use innovative techniques and tools in computing practices, fulfilling the outcome related to software engineering and design.

In terms of outcome coverage, I believe I have met and exceeded the goals I set in Module One. My original plan was to demonstrate mastery in algorithmic design, recursive logic, and test-driven development. Through the enhancement process, I expanded my scope to include performance tuning, UI integration, and coroutine compatibility. These additions reflect a deeper understanding of the trade-offs involved in software design, such as balancing readability with efficiency and ensuring that data structures remain robust under real-world usage scenarios. I also gained insight into the importance of modular architecture and dependency injections, which allowed me to isolate the BST logic from Android-specific dependencies and ensure JVM-safe unit testing. This approach aligns with best practices in software engineering and supports the development of maintainable, scalable systems.

Reflecting on the process of enhancing and modifying the artifact, I encountered several challenges that required creative problem-solving and technical rigor. One of the most significant challenges was adapting recursive traversal logic to work within Android’s threading model, which necessitated the use of coroutine scopes and careful synchronization. I also faced difficulties in ensuring that the comparator logic remained flexible yet type-safe, particularly when sorting by different fields with varying data types. Through these challenges, I learned the value of abstraction and interface-driven design, which allowed me to decouple sorting behavior from the core BST implementation. Additionally, I deepened my understanding of unit testing frameworks such as JUnit and MockK, which enabled me to write comprehensive tests that validated both expected and edge-case behavior. These experiences reinforced the importance of iterative development and continuous testing, which are essential for producing reliable software.

Scholarly literature supports the importance of mastering data structures and algorithms in professional software development. As Weiss (2014) notes, “The choice of data structure and algorithm can have a profound impact on the performance and scalability of a system” (p. 3). This insight is particularly relevant to my artifact, where the BST enables logarithmic-time operations for insertion, deletion, and search, critical for managing large inventories efficiently. Weiss also emphasizes the importance of understanding the trade-offs between different design choices, such as time complexity versus memory usage, which I encountered firsthand while optimizing the BST traversal logic. By applying these principles in a real-world context, I demonstrated not only technical proficiency but also the ability to make informed design decisions that enhance the overall quality of the application.

In conclusion, Milestone Three allowed me to take a significant step toward demonstrating competency in algorithms and data structures, as well as broader software engineering practices. The artifact I selected and enhanced reflects a deep engagement with core computer science principles and showcases my ability to design, implement, and refine complex systems. The accompanying narrative provides a clear and technically sound explanation of my process, learning outcomes, and challenges faced. Together, the artifact and narrative illustrate substantial progress toward final project readiness and provide a strong foundation for my professional ePortfolio. I look forward to incorporating instructor feedback and further polishing the artifact to ensure it reflects the highest standards of quality and professionalism.

*Reference:*  
Weiss, M. A. (2014). *Data Structures and Algorithm Analysis in Java* (3rd ed.). Pearson Education. Retrieved from Shapiro Library