**Algorithms and Data Structure Narrative**  
By Wilfredo Mendez

This artifact is my final project submission for CS 320: Software Testing, Automation, and Quality Assurance, created to demonstrate my abilities in software testing and data structure validation. It originally included four files; two define the Appointment class with its associated getters and setters, and an ArrayList class that holds the appointments, along with CRUD operations and validation tests. The other two are test files; one testing the Appointment class boundaries and methods, and the other testing the ArrayList's boundaries and validations. The original goal of this project was to assess my ability to build comprehensive unit tests that ensure code correctness and quality.I selected this project because it showcases my understanding of data structures and testing practices, as well as my ability to optimize software performance.

For this enhancement, I introduced a new version of the dataset using a HashMap and conducted comparative tests between the ArrayList-based solution and the HashMap-based solution. The primary focus of these tests was to demonstrate how different data structures affect performance. I highlighted the time complexity differences between the two implementations, showing that HashMap, with an average-case complexity of O(1) for insertion and retrieval, offers a significant performance improvement over the ArrayList, which has O(n) complexity for search and deletion operations. I also visually demonstrate the speed differences in the data structures to show their time complexity and visualize how changing data structures can make the code more optimal, showcasing my understanding of more efficient data structures.

Through this process, I learned to think critically about design trade-offs between data structures. While the ArrayList provided predictable ordering, the HashMap delivered superior performance for the operations most frequently used in this application. This enhancement allowed me to reflect on real-world development scenarios where performance and usability must be balanced to meet application needs. I also modified the test cases to ensure the new structure’s boundaries and components were thoroughly tested, demonstrating my ability to adapt and test diverse data structures.

Overall, this enhancement demonstrates my ability to design and evaluate computing solutions, apply innovative tools, and foster collaborative coding practices. The switch from ArrayList to HashMap highlights my understanding of data structure trade-offs, demonstrating course outcome three. The improved testing framework demonstrates my growth in applying innovative testing tools, and the comparison demonstrates outcome four.  
  
Outcome Three: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices. This outcome was achieved through the evaluation of trade-offs between ArrayList and HashMap for storing and retrieving appointments. ArrayList’s ordered nature was useful in the original implementation, but HashMap provided better performance for the type of operations needed—especially retrieval and deletion. I ensured the correctness and efficiency of the new solution through timing tests and documented my decisions in the code comments to reflect a deep understanding of algorithmic choices and data structure optimization.

Outcome Four: Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. This outcome was met by incorporating nanoTime-based performance testing into the code. I had limited experience with precise performance measurement tools before this project, but I learned how to use them effectively to assess the efficiency of the new structure. I also modified the test cases to thoroughly evaluate both the HashMap and ArrayList, ensuring their boundaries were respected. This use of innovative testing techniques highlights my ability to apply appropriate tools and strategies to create value through optimized code.