# CS499 Artifact Two Narrative

In one of the previous assignments at Study.com during the Computer Science 201 - Data Structures & Algorithms course, we were tasked with developing a Java application that sorts an unsorted list of States and Capitals in natural order (alphabetical), either by State or by Capital. Additionally, we needed to create a HashMap (to demonstrate that it does not guarantee the order) and a TreeMap (to show that it sorts in natural order by default) of States and capitals. Finally, a Trivia Game to test knowledge of states and their capitals needed to be implemented, along with a single question/answer module to determine the capital of a particular state.

I selected this artifact to demonstrate my knowledge and understanding of data structures and algorithms, as well as my ability to develop creative solutions for various problems in the computer science field. From arrays, hash maps, and treemaps to the bubble sort algorithm and its optimizations, my artifact encompasses a wide range of topics in algorithms and data structures, utilizing Java as the primary programming language. The enhancements I implemented include optimizing the bubble sort functions bubbleSortCapsFunc(myArray) and bubbleSortStatesFunc(myArray) to sort a 2D array of capitals and their corresponding states, either by state or by capital, preventing the code from continuing to iterate through the array even after it was already sorted. The optimized versions are, on average, 20% more efficient because they require around 20% fewer iterations. This was demonstrated with the newly developed method getAverageOptimization(iterations), which creates two identical arrays, sorts both the optimized and non-optimized versions using a bubble sort, given the number of iterations, and calculates the average improvement as a percentage. Another improvement was to add save-score functionality for the Trivia Game of States and Capitals, where I created an algorithm to save the user’s name and scores to a file and display the scoreboard, sorted from highest to lowest, using the bubble sort algorithm. Finally, a main menu was added for users to pick one of the following options: print unsorted array of states and capitals, sort by capitals by using the optimized bubble sort method, sort by states by using the optimized bubble sort method, play Trivia Game, a single Q/A Game to practice knowledge of states and their capitals, print optimization in percentage based on interations, print our Score Board for Trivia Game, and quit application.

By implementing my enhancements, I achieved both of my planned course outcomes, which demonstrates my ability to 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, as well as 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. My outcome coverage plan is still the same as it was at the beginning of the course.

I got stuck a couple of times during my enhancement process. The first challenge was related to arrays, as I had to refresh my knowledge on how Java initializes them, their available methods, and how to clone them properly. I had to create an additional method, getCloneOf2D(myArray), that returns a manually cloned 2D array, since the included methods clone() and copyOfArray() create only shallow copies, which essentially create a reference to the original. The second one was creating an algorithm that saves users’ names and scores in a file. I had to add a first line to the file that includes the number of users in the scoreboard and update it every time a user is added, so I can create an array of that dimension to perform the bubble sort algorithm and display the results from highest to lowest in the user's console using the showScoreBoard() method. I learned several valuable techniques along the way that helped me become a better software engineer.