

The artifact being presented for enhancement two was my final project in CS410 Software Reverse Engineering during 21EW4 and is named Loginapplication.cpp. The idea behind the project was to create an application from reverse software engineering an application and then through the interpretation of the assembly language to construct the application utilizing C++.

This artifact was selected for the enhancement of algorithms and data structures due to the lack of a proper login credentials matching function within the application. Thus, the need for a data structure as well as algorithms within this artifact seemed the most apparent to accommodating a couple course objectives. Within this artifact I created a hash table data structure to accommodate this function of matching login credentials as well as accompanying Hash Table functions that utilize “for” loops and pointers for Hash table traversal allowing for insertion, deletion, searching, and display of the values within the table. The inclusion of this hash table provides the artifact with a data structure that stores user numbers (keys) and passwords(values) within  $O(1)$  time complexity if a proper hash function is selected. This allows for a storage of data that is retrieved at the highest speed. The creation of this hash table displays my ability to not only create an adequate data structure for the project but also displays an understanding of the need for this structure to exist within the project. Beyond the creation of this data structure, a nested switch case was utilized to accommodate the use of the hash table functions and provide an administrator functionality of adding, removing, and displaying users within the artifact. Furthermore, the inclusion of the search function algorithm within the artifact allows for a verification of user credentials before access to the general menu is provided to the user. Specifically, the search hash table function in cooperation with a while loop serves to exemplify not only my abilities regarding the creation of the data structure required but also the

application of the function to serve a purpose within the overall artifact. Other algorithms including if, for, and while loops as well as switch cases were utilized within the artifact as well. These functions and algorithms as well as the creation of the hash table serve to improve the artifact overall by accommodating not only a credential checking functionality but also administrator functionality within the project.

In the creation of this hash table and algorithms to produce an overall improvement within the artifact a couple course objectives were achieved. The objective defined in the course description as “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 (data structures and algorithms)” was addressed. This objective was addressed through the creation of the hash table as well as its accommodating algorithmic functions described previously to provide an appropriate solution within the project to answer the question of user credentials verification. Secondly the course objective described as “Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources” was also achieved using closed off loops as well as input validations (cin. clear, cin. Ignore etc.) to create functionality while keeping a security mindset to ward against potential exploitations. In this artifact enhancement both course objectives were effectively addressed and met to proficiency through the provided source code implemented.

In the creation of this enhancement, I learned the importance of data structure selection. Originally the idea was to implement an array with user information but during my initial attempts it became apparent that the immutable nature of the array was problematic in user

credential storage. A hash table seemed an adequate solution to the expected result of storage structure that could store user information storage therefore the implementation of the hash table provided not only the storage but also the efficiency of data retrieval not provided to other data structures. During the initial implementation, I found challenges in producing an adequate search function. Upon realization that I was not returning the value upon discovery the search function was corrected and functional. A value lesson that it is important to acknowledge what you are returning in a function. In creating a menu to implement administrator functions I was met with a repeating output of menu functions resulting in a frustrating application. This led me to another learning realization regarding the importance of break calls. The nested switch case did not have an individual break call resulting in a never-ending output of admin menu. These challenges required critical thinking and analysis to address the issues thus resulting in adequate solutions.