Jonathan Klein

Mr. Watson

AP Computer Science A

15 October 2019

Loops Project Outline

To begin this project, I knew that I began by creating a simple password generator of length 10 with letters, numbers, and special characters. To do this, I made a string of all the characters I wanted in my password. Then, I created a for loop that repeated 10 times and concatenated a random character from the string into a new string, which I made to store the key. Then I created a scanner object to take in user input, alongside a variable to store the amount of attempts. Since the user would have unlimited attempts, I made the while loop depend on whether the password was cracked or not, a boolean variable. The rest was rather straightforward. I continued by updating the user input variable, and then I checked whether it matched the key generated previously. If they matched, the while loop boolean was changed and the while loop was therefore terminated. If not, the loop continues. I then added my quit method for the user, which the user achieves by typing quit into the input. I did this by simply checking user input for ‘quit’, and then used the break; method to terminate the while loop. The only thing that I needed to add at this point was the conditions that make the password reset, and increase in length by one character. This was to be done when a common password was entered, and every multiple of 10 of attempts. To do this, I created an ArrayList with all the common passwords, and searched them. After every time the user entered something, it was checked whether his/her input was in the ArrayList of common passwords, or if the amount of attempts was a multiple of 10. If either was true, a new password was generated with one length more than what it was originally. This was the base code for the requirements, and I polished it by adding implementing ascii art and line methods to neaten up the UI.