**Phase 4 Grading Criteria**

* Test and revise the application to do the required tasks stated in the first three phases.
* if you are going to use two screens, how would you do it? The new screen displays the remembered phrases entered from the first screen. Implement the application to do this task.

**Detailed grading criteria for the final project (Phase 4). The following information will be emailed to you as well.**

* 1. (5 points) Copy and paste every line of code in the project into a Word document in which the classes and/or interfaces that manage passwords and scores will be displayed first. These two classes are not one that extends Application or JPanel or JFrame. Then copy the classes that contain GUI code components followed by the main ( ).

       (5 points) UML class diagrams that align with the code implementations.

* 1. (10 points) Highlight the lines of code that demonstrate a class relationship (“is a “and/or “has a “relationship) between two classes and/or interfaces. Below the code as a comment, **use dark green text for an explanation about the class relationship. Explain if your UML class diagrams align with the implementations. Also, state if there is anything you would have done better and any code duplications in these classes that could be revised.**
  2. (10 points) Highlight the lines of code that demonstrate where you declare and instantiate the objects of the classes that manage data (passwords and scores/games). Below the code as a comment, use blue text for an explanation about the required tasks and how they work.
  3. (10 points) Highlight the lines of code that demonstrate where you save the information on disk.  Below the code as a comment, use brown text for an explanation about the required tasks and how they work. Is the file access binary and random access or a database and  did you retrieve the information from disk?

     (5 points) files on disk- must be submitted.

* 1. (10 points) Highlight the lines of code that show class methods that create strong passwords. Below the code as a comment, use blue text for an explanation where string inputs come from and how they are manipulated.
  2. (10 points) Highlight the lines of code that show GUI for a game-play (as supposed to be done in phase 1). Below the code as a comment, use red text for an explanation where string inputs come from and how they are manipulated.
  3. (10 points) Highlight the lines of code that show GUI for string selections used to pass to class methods for strong password creation (as supposed to be done in phase 1). Below the code as a comment, use purple text for an explanation where string inputs come from and how they are manipulated.
  4. (10 points) Highlight the lines of code that show the second window and its implementation in Phase 4.
  5. (5 points) **Provide two screenshots of outputs** that show results of a game-play.
  6. (5 points) **Provide two screenshots of outputs** that show results of password created.

**What to turn in**

**The word document file created, text files, UML class diagrams, and screenshots of the outputs as requested in #9 and #10.  Please do not zip files submitted.**