Journal Design

Tracy Mann

CSE210

Bro. Parrish

Section A8

5/20/2023

v1.0

# What does the program do?

The program will display a full scripture and then hide a few words at a time until the complete scripture is hidden.

## Functional Requirements:

This program must contain the following features:

1. Store a scripture, including both the reference (for example "John 3:16") and the text of the scripture.
2. Accommodate scriptures with multiple verses, such as "Proverbs 3:5-6".
3. Clear the console screen and display the complete scripture, including the reference and the text.
4. Prompt the user to press the enter key or type quit.
5. If the user types quit, the program should end.
6. If the user presses the enter key (without typing quit), the program should hide a few random words in the scripture, clear the console screen, and display the scripture again.
7. The program should continue prompting the user and hiding more words until all words in the scripture are hidden.
8. When all words in the scripture are hidden, the program should end.
9. When selecting the random words to hide, for the core requirements, you can select any word at random, even if the word was already hidden. (As a stretch challenge, try to randomly select from only those words that are not already hidden.)

## Design Requirements

In addition, the program must:

1. Use the principles of Encapsulation, including proper use of classes, methods, public/private access modifiers, and follow good style throughout.
2. Contain at least 3 classes in addition to the **Program** class: one for the scripture itself, one for the reference (for example "John 3:16"), and to represent a word in the scripture.
3. Provide multiple constructors for the scripture reference to handle the case of a single verse and a verse range ("Proverbs 3:5" or "Proverbs 3:5-6").

## Stretch Requirements

1. The program works with a library of scriptures rather than a single one. Choose scriptures at random to present to the user.
2. The program loads the scripture library from a file.
3. The program will track and save when a word is hidden in the file. This will be reset once the user presses enter after the last word is hidden in the scripture.
4. The program will track the number of times a scripture is used, this will be used to balance the distribution of the random selection of a scripture to be shown. Weighted towards least used scriptures.

# What user inputs does it have?

1. Scripture action prompt response.

# What output does it produce?

1. display scripture.

{book} {chapter}: {verse number} {verse text}

…

{verse number} {verse text}

* 1. note: when a word is hidden, characters previous to the first each alpha numeric character and any characters after the last alpha numeric character will still be shown. The rest will be replaced by an underscore.

1. Scripture action prompt.

Press enter to continue or type ‘quit’ to finish:

>

1. exit message

Thank you for using the Scripture Memorizing Program, hope to see you soon.

# How does the program end?

# The user will either press enter after the last word is hidden or type quite at any prompt during the program’s operation.

# Class Diagram:

