**Scripture Memorizer Program Design**

1. What does the program do?

The program displays a full scripture and then hides a few words in the scripture at a time, continues to hide more words until all words in the scripture are hidden or the user types ‘quit’.

1. What user inputs does it have?

* ‘Enter’ keystroke
* Quit

1. What output does it produce?  
    - Full text of scripture and its reference

- Text of scripture with random words hidden

1. How does the program end?

- User types ‘quit’

- When all words in the scripture are hidden

**Classes**

* **Scripture**

Responsibilities:

* Keeps track of scripture reference
* Keeps track of each word in the Scripture

Behaviours:

* Can hide word
* Can tell if all the words in a scripture have been hidden
* Gets refence and scripture text in the formatted manner for display
* **Word**

Responsibilities:

* Keeps track of a word
* Tracks the visibility state of the scripture
* Can hide word
* Can return the word as text

Behaviours:

* Tells if a word is hidden or not
* Can hide a word
* Can return the literal text of a word
* **Reference**

Responsibilities:

* Keeps track of a book name/title
* Keeps track of a chapter in that book
* Keeps track of a starting verse in that chapter
* Keeps track of ending verse in that chapter

Behaviours

* Returns a formatted reference in book chapter:startverse-endingverse format
* **File**

Responsibilities:

* Keeps track of scripture text and references.

Behaviours:

* Can read the text and references of a scripture from a file.
* Can generate a random scripture.
* Can output a list containing all references to the scriptures stored in a file.
* Can display a scripture selected by the user.
* **ScritpureAPI**

Responsibilities:

* Keeps track of scripture reference.
* A dictionary that keeps track of scripture references and text gotten from an API endpoint.

Behaviours:

* Constructor that accepts a reference as an argument.
* Queries an external API endpoint using the reference passed to the constructor.
* Returns the text and reference gotten from the API endpoint to the user as a dictionary.

