- **Developer**: Nalin

Design of `constructor()`

- Define a constructor `init` with `self` being the current instance of the class.
- The constructor `init` has one parameter `scrabble_file` of datatype `file`.
- Initialize the instance variable `self.scrabble_letter` with an empty dictionary.
- Define a local variable `letters_file` of the datatype `file`.
- Use the local variable `letters_file` to open the file given as the parameter `scrabble_file` in read mode by encoding it.
- Define a local variable `list_of_lines` hwose datatype is a `list`.
- Use the `readlines` method on `letters_file` and assign it to the variable `list_of_lines`.
- Use `for` loop to iterate through `list_of_lines` with `line` as the iterator variable whose datatype is `string`
    - Skip the first element in the list as it is "letter,count,points" by using.
    - Inside the `for` loop define three more variables `letter`, `frequency`, `points` of datatype `stinrg`.
    - Use split method on each `line` with `,` as the separator and assign it to the three variables `letter`, `frequency`, `points`.
    - If `letter` is the string `"balnk"` then
        - Assign an string with a blank space to `letter`.
    - Assign the list of `frequency` and `count`, where their datatypes ae typecasted to `int`, to the key `letter` in the dictionary `self.scribble_letters`.

Design of `reduce_frequency()`

- Define a method `reduce_frequency` with two parameters.
- The first parameter being `self` of datatype class instance and the second being `letter` of datatype `string`.
- If the `letter` is a string called `blank` then assign a string containing a blank space to `letter`.
- Update the string `letter` by using `.lower()` method on it.
- If the `letter` is in the dictionary `self.scrabble_letters` then
    - Define the local variables `frequency`, `points` of datatype `int`.
    - Assign the `value` of the `key`, `letter` in the dictionary `self.scrabble_letters` to the local variables `frequency`, `points`.
    - If the `frequency` is greater than 0 then
        - Reduce the value of the first element in the `list` associated as `value` to the `key`, `letter` in the dictionary `self.scrabble_letters`.
        - Print the updated dictionary `self.scrabble_letters`.
        - Return `True`.
    - Return `False`.

Design of `get_freq()`

- Define a method `get_freq` with two parameters.
- The first parameter being `self` of datatype class instance and the second being `letters` of datatype `string`.
- Initialize the local variable `count` with an empty dictionary.

- Use the `for` loop to iterate through the sring `letters` with the iterator variable as `letter` of datatype `string`.
    - Update the letter by converting it into lower case using `lower()` method.
    - If the `letter` is in the dictionary `self.scrabbe_letters` then
        - Assign the first element in the list, associated to the `key`, `letter` in the dictionary `self.scrabble_letters` to the `key`, `letter` in the dictionary `count`.
- Return the dictionary `count`.

Design of `get_points()`

- Define a method `get_points()` with two parameters.
- The first parameter being `self` of datatype class instance and the second being `word` of datatype `string`.
- Initialize the local variable `points` of datatype `int` with 0.
- Use `for` loop to iterate through the srting `word` with the iterator variable `letter` of datatype `string`.
    - Update the letter by converting it into lower case using `lower()` method.
    - If `letter` is in the dictionary `self.scrabble_letters` then
        - Add the second element in the list, associated to the `key`, `letter` in the dictionary `self.scrabble_letters` to the variable `points` and assign this to `points`. Return `points`.