Assignment 2

Word Game

**Due:** Tuesday, November 7th

**Objectives:** Input/Output, Variables, Strings, Chars, Methods

This assignment is to be done individually.

For this assignment we will be creating a two-player word guessing game. To play, one player chooses a secret word to be guessed. Then, the next player will attempt to guess the word. After each guess, the computer will tell the player how many of letters in the guessed word are in the secret word and how many of the letters in the guessed word are in the correct locations in the secret word. Using this information, the guessing player will guess again, getting more information until they figure out the correct answer. A full implementation of this game can be found at <https://word500.com/>, with a more detailed explanation at <https://word500.com/helpen>.   
  
Play continues until a player finally guesses all letters in the word, in which case they win, or until they make 12 incorrect guesses, in which case they lose. To make programming this a bit easier, our version of the game will have a couple of restrictions: The secret word and the guessed words can only be 5 characters long, all lowercase. Additionally, these words cannot have repeat characters (IE: “deter” would not be a valid word because it has 2 e’s)

# General considerations:

Your code must follow the structure detailed below. It is important to test your methods, you will be required to create and submit at least 3 test cases for **isLetterInWord** and **printWord**. You will be evaluated on how well you follow best practices when programming. Your code must be properly indented, contain good, descriptive variable names, have helpful inline comments, and ***use constants where appropriate***.

Your program should be created in a class called WordGame, with the following methods:

**boolean isLetterInPlace(String word, char letter, int position):**

You should write a method isLetterInPlace that takes as parameters a String representing a word, and a char containing a character we would like to find in that word, and an int representing the position we are checking for that letter. The method should return true if the character at the specified position matches the specified letter, and false otherwise.

Note: The only String method you should use for this part is the charAt() method to get characters from the String one at a time. Using other methods will cause you to lose marks!

For example:

If the parameters are "both", ‘t’, **2** return true

If the parameters are "both", ‘t’, 1 return false

**boolean isLetterInWord(String word, char c):**

You should write a method isLetterInWord that takes as parameters a String representing a word, and a char containing a character we would like to find in that word. The method should return true if the character exists in the word, or false if it doesn’t.

Note: The only String methods you should use for this part is the charAt() method to get characters from the String one at a time. Using other methods will cause you to lose marks!

For example

If the parameters are "both", ‘t’ return true

If the parameters are "both", ‘a’ return false **Requirement:** If you find characters individually by using repetitive/copy pasted code, you will not get full marks on this question. For the highest possible grade, you should find the character using a loop. (Suggestion: Solve it first using whatever means is easiest for you, and then try to fix it up using a loop afterwards)

**int** **countLettersInPlace(String answer, String guess):**

You should write a method countLettersInPlace that takes as parameters a String representing an answer, and String representing a guess. Using the method isLetterInPlace written earlier, you should count the number of letters in the guess which are in the same position in the answer. The method should return the number of letters which are in the same positions in the two words.

Note: The only String methods you should use for this part is the charAt() method to get characters from the String one at a time. Using other methods will cause you to lose marks!

For example

If the parameters are "both", “bath” return 3

If the parameters are "both", “pods’ return 1

If the parameters are "both", “thus’ return 0 **Requirement:** If you solve this problem using repetitive/copy pasted code, you will not get full marks on this question. For the highest possible grade, you should solve it using a loop. (Suggestion: Solve it first using whatever means is easiest for you, and then try to fix it up using a loop afterwards)

**int countLettersInWord(String answer, String guess):**

You should write a method countLettersInWord that takes as parameters a String representing an answer, and String representing a guess. Using the method isLetterInWord written earlier, you should count the number of letters in the guess which are in the word (not necessarily the same position). The method should return the number of letters which a present in both words.

Note: The only String methods you should use for this part is the charAt() method to get characters from the String one at a time. Using other methods will cause you to lose marks!

For example

If the parameters are "chip", “path” return 2

If the parameters are "chip", “clot” return 1

If the parameters are "chip", “stun” return 0 **Requirement:** If you solve this problem using repetitive/copy pasted code, you will not get full marks on this question. For the highest possible grade, you should solve it using a loop. (Suggestion: Solve it first using whatever means is easiest for you, and then try to fix it up using a loop afterwards)

# boolean assessWord (String answer, String guess)

Here we start using our methods to put together our game. You should write a method called assess word that takes a String representing an answer and a string representing a guess. Using the methods defined above, your method should print out the number of letters in the same place in the guess as in the answer, as well as print out the number of letters that are in the guess as well as the answer (not in the same place). If all of the letters match, and the words are the same, return true, otherwise return false.

# void runGame(String answer)

You should write a method **runGame** which will run the game. This method should take as a parameter the answer that the user is trying to guess.  
  
Using a loop, you should give the user up to 12 attempts to guess the word. After each guess, you should use the assessWord method to give the user feedback on how close their word was to the answer. If the user successfully guesses the word before their 12 attempts are up, you should print a message saying that they got the word, and congratulating them on winning the game. If the user made 12 guesses and did not correctly guess the word, you should print a message saying “you lose” and admonish them for their failure.

**Note: While testing this method, it’ll be easier if the amount of loop iterations you have to go through to see a losing message is lower than 12. Make the amount of attempts that the user has easily modifiable, and while testing, set it to 2 or 3.**

**void main(String[] args)**

Your main method for this program should be doing very little. All you should do in your main method is use a Scanner to request a word from the user, and then use that word to call the runGame method and start the game.

# To submit:

* Submit WordGame.java on moodle.
* The test cases file (testcases.txt) to submit on moodle:

Create a file using notepad++ for methods isLetterInWord() and countLettersInWord(), and provide at lease three independent test cases that you used to test each of your methods, all parameters and expected results. Your test cases should be distinct from the examples presented in this document.

A test case would look like the following:

* + isLetterInWord("both", ‘t’) return true
  + countLettersInWord(“both”,”thud”) return 2