

Programming Assignment 2

GUESS THE WORD

Objectives:

- Practice if and if/else statements

Assignment:

For this assignment, create a simple word game. The user must guess a word consisting of four letters. The first and last letters are already given. The user guesses one additional letter, and then guesses the entire word.

WordGame Class

This class will play the game. You can pick your own word. All four letters must be unique.

First, display the information about the game. Display the word with the first and last letters filled in and ask the user for a letter. Then, check if the letter is in the word. If it is, update the word being displayed to include the letter in the correct position. If it's not, display the word again with just the first and last letters. Next, ask the user to guess what the whole word is. Finally, determine if the user correctly guessed the word, and display whether they won or lost the game.

Test Cases

As you write your program, you will need to test it, especially your decision structures. Download the Test Cases file from Moodle and fill in each row with the tests you run.

Sample Execution:

Sample Execution 1:

```
*****
*           GUESS THE WORD           *
*****
*
* The word has four different *
* letters. The first and last *
* are filled in. You have one *
* chance to guess one other *
* letter the word. Then, you *
* will guess the full word.    *
*
*****
```

```
=====
=====| GUESS  1 |=====
=====
```

```
+-----+
|  f__e  |
+-----+
```

Enter a lowercase letter: **i**

```
=====
=====| GUESS WORD |=====
=====
```

```
+-----+
|  fi_e  |
+-----+
```

Enter the word: **five**

```
=====
```

```
*****
You Win!!!
*****
```

```
+-----+
|  five  |
+-----+
```

```
=====
```

Sample Execution 2:

```
*****
*           GUESS THE WORD           *
*****
*                                     *
* The word has four different        *
* letters. The first and last        *
* are filled in. You have one        *
* chance to guess one other          *
* letter the word. Then, you         *
* will guess the full word.          *
*                                     *
*****
```

```
=====
=====| GUESS  1 |=====
=====
```

```
+-----+
|  f__e  |
+-----+
```

Enter a lowercase letter: v

```
=====
=====| GUESS WORD |=====
=====
```

```
+-----+
|  f_ve  |
+-----+
```

Enter the word: a

```
=====
```

```
*****
You Lose!!
*****
```

```
+-----+
|  five  |
+-----+
```

```
=====
```

Sample Execution 3:

```
*****
*           GUESS THE WORD           *
*****
*                                     *
* The word has four different        *
* letters. The first and last       *
* are filled in. You have one       *
* chance to guess one other         *
* letter the word. Then, you        *
* will guess the full word.         *
*                                     *
*****
```

```
=====
=====| GUESS  1 |=====
=====
```

```
+-----+
|  f__e  |
+-----+
```

Enter a lowercase letter: a

```
=====
=====| GUESS WORD |=====
=====
```

```
+-----+
|  f__e  |
+-----+
```

Enter the word: fire

```
=====
```

```
*****
You Lose!!
*****
```

```
+-----+
|  five  |
+-----+
```

```
=====
```

Requirements:

- Use an updated comment block
- Your program should use the following comment block at the very beginning of your program.

```
// Name: Your Name                      Date Assigned: Fill in
//
// Course: CSCI 2003 42733              Date Due: Fill in
//
// Instructor: Ms. Greer
//
// File name: Fill in
//
// Program Description: Brief description of what the program does.
```

- Use appropriate comments throughout the program
- Make good use of whitespace
- Your output should be neat and easy to understand.

Deliverables:

- WordGame.java file
- Test Cases Excel file
- Upload both files to Moodle

Grading:

Total Points	15 points
WordGame Class	13 points
Create full word for answer	1 point
Create word with blanks for display	1 point
Display game info	1 point
Display word with blanks	1 point
Get letter from user	1 point
Determine if letter is in word	4 points
Determine if letter is in first blank	1 point
If yes, replace first blank with the letter	1 point
Determine if letter is in second blank	1 point
If yes, replace second blank with the letter	1 point
Get word from user	1 point
Determine if word is correct	3 points
If yes, display You Win!!!	1 point
If no, display You Lose!!	1 point
Display correct word whether user wins or loses	1 point
Test Cases	2 points
Not enough comments/whitespace	-1 point
Output is messy / difficult to read or follow	-1 point
Bad variable names, method names, and/or class names	-1 point