If two words have a letter in common but the letter appears in the words a different amount of times (i.e. three m’s in ‘mommy’ and two m’s in ‘comma’), how many times should I count that letter as being in common?

Once a particular letter is counted as in common, it cannot be counted again. Consequently, there are two m’s in common in the example above. Another example: happy vs. sappy – 4 letters in common.

When should the game display dialog messages to the user?

The game should display messages indicating that the user has won or lost the game when the game ends and to indicate any errors.

My game takes 15+ guesses to get my word. Is this okay?

Generally, it should not take 10 guesses for the computer to guess the word. If your game is consistently taking longer, there is probably something wrong with the way your code eliminates words.

How do I make a Random object?

You can create a Random object by calling its constructor. You can do so by:

```
Random varName = new Random();
```

Use the Java APIs and look at the code from class to figure out how to use the Random object.

What makes my code robust?

The only check you need to make is if the word is not in the list or if the user inputs inconsistent numbers. These would both lead to the same error message at the end of the game saying the computer could not find the word.

Should I write code to end the game after 15 guesses?

Yes. While a given game normally should not take that long, you should write code such that if there are ever 15 turns, the computer loses.

Let’s say the computer guesses a word that has 5 letters in common with the word I am thinking of - for example, it guesses melba and my word is amble. Should the word the computer guesses be removed from the list for the next turn?
Yes, the word that the computer guesses should *always* be removed from the list before the next turn. If the computer guesses wrong, even if there are 5 letters in common, you would not want the computer to make this incorrect guess again.