CIT 149: Java I
Chapter 5 Programming Assignment 2

This is a more difficult assignment so I am giving you some steps to completing it.

You will need the following in this program:

- Variables for:
  - String: *secretWord*
  - String: *disguisedWord*
  - String: *lettersRemaining*
  - Integer: *guessesMade*
  - Integer: *incorrectGuesses*
- Create a method named initialized(). This method should have parameters of a single String of *word*. Within this method:
  - set *secretWord* to equal the String variable that is part of the method parameters, changing it to lowercase and trimming it of any white space. For example if your header was:

    public void initialize(String word)

    you would have:

    secretWord = word.toLowerCase().trim();

  - Set the variable *lettersRemaining* to the value of *secretWord*
  - Set the variable *disguisedWord* to equal what is returned by a method named createDisguisedWord() with an argument of the *secretWord* variable.
  - Set the variable *guessesMade* and *incorrectGuesses* to zero.

- Create a method named createDisguisedWord with parameters of a String named *word*. This method will return a String so instead of void in the method header you will have String. This method has very simple, repeated code. I will give you the first two lines:

  word = word.toLowerCase();
  word = word.replace('a', '?');

  Continue this for every letter of the alphabet.

At the end of the method have the method return *word*.

- A method named makeGuess() which has parameters of a Character named *c*. Within this method:
    - create an if statement. This if statement will check to see if Character is the letter c, as in:

      if(Character.isLetter(c))

    - This uses the Character class' isLetter method to check to see if the letter to be guessed is the letter passed to the method.
    - Within the if statement:
        - Create a String named *guess* and have it equal an empty String plus c as in:

          String guess = "" + c;

        - Change *guess* to lowercase as done in the previous method.
        - Create an integer named *letterPosition* that equals the value of the variable *lettersRemaining* with an index number of *guess*. As in:

          int letterPosition = lettersRemaining.indexOf(guess);

        - Create a boolean named *goodGuess* that equals *letterPosition* > -1. This will set to true if the condition is true, false otherwise.
        - Create a while loop with the argument of *letterPosition* > -1. In the loop:
            - Create a String named *before* that equals, lettersRemaining.substring(0, letterPosition)

            - Create a String named *after* that equals: lettersRemaining.substring(letterPosition+1)

            - set the variable *lettersRemaining* to equal: before + "#" + after
            - set the variable *before* to equal: disguisedWord.substring(0, letterPosition);
            - set the variable *after* to equal: disguisedWord.substring(letterPosition+1)
            - set the variable *disguisedWord* to equal: before + guess + after
            - set the variable *letterPosition* to equal: lettersRemaining.indexOf(guess)

        - Close the loop.
            - Increase the value of *guessesMade* by 1.
            - Create a nested if statement of:

```
if(!goodGuess)
    incorrectGuesses++;
```

    - Close the first if statement.
    - Create a else statement that displays a message stating that "Sorry, your guess must be an alphabet character from a to z"
- Close the method.
- Create a getDisguisedWord() method that has no parameters, and will return a String data type. Within the method have it return *disguisedWord*
- Create a getSecretWord() method that has no parameters, and will return a String data type. Within the method have it return *secretWord*
- Create a getGuessCount() method that has no parameters, and will return an integer data type. Within the method have it return *guessesMade*.
- Create a isFound() method that has no parameters, and will return a boolean data type. Within the method have it return *secretWord.equals(disguisedWord)*
- Create a playGame() method that has no parameters and is void. So it will not return anything. Within this method:
    - Create a if statement based on what is returned by the isFound() method being false as in: if(!isFound()). Within this if statement:
        - Create the code that will display "We are playing hangman"
        - Create while loop with the same argument as the if statement. Within the loop:
            - Type the code that will display "The disguised word is <" plus the value of the variable *disguisedWord* plus ">"
            - Type the code that will display "Guess a letter"
            - Create a Scanner object named *keyboard*
            - Create a String variable named guess and have it equals what the user enters as in: String guess = keyboard.next();
            - Create an if statement based on guess.length() !=1. Within this if statement have it display "Sorry, bad guess. Need a single letter"
            - Close the if statement if you included curly braces.
            - Create the else statement which will invoke the makeGuess() method passing to it a new Character for the guess at character zero, as in:

makeGuess(new Character(guess.charAt(0)));

- Close the else statement.
- Type the code that will display "Guesses made " plus the value of the variable *guessesMade* plus " with " plus the value of the variable *incorrectGuesses* plus " wrong"
- Close the loop
- Type the code that will display "Congratulations, you found the secret word: plus the value of the variable *secretWord*

- Close the if statement.

- Close the method
- Create your main method. Within the main method:
  - Create a Hangman object named *game*.
  - Call the initialize method for the object with an argument of "Happiness" as in:

    game.initialize("Happiness");

  - Write the code that will display "Lets play a round of hangman"
  - Invoke the object's playGame() method.
  - Invoke the object's initialize() method again but this time with the parameters of "I'll be back"
  - Type the code that will display "Lets play a second round of hangman"
  - Invoke the object's playGame() method again.
- Close the main method and the class.
- Compile the program, fix any errors and run the program.