Michael Smith

Step 1:
1. The user inputs would consist of the letter that they guess. If you wanted, you could involve a user input that is the word needed to be guessed. If you did that, you could have two players and one gets to choose the word.
2. According to the example, the outputs would be the amount of guess' you have left, the letters guessed correctly in the word, and the letters guessed incorrectly. Other outputs would consist of the beginning and end of the game, stating whether or not you lost or won.
3. We are assuming that the user inputs consist of letters. You can also assume that if the user entered a number that the program would ask again.
4. The tasks of the program would be repeating the guessing process until the game is over and showing the letters already guessed and the number of tries left. The subtasks would be assigning each letter of the word to a variable and seeing if any guess' are correct or not.

Step 2:
1. The input from the user should be stored into a variable and then returned out of the guessing function so it's able to be used outside of it.
2. The word is also stored in a string variable but each letter of the word should also be stored in a string variable because the user isn't guessing words, they're guessing letters.
3. You only read inputs from the user when they guess a letter. The code needs to store the guess and then analyze that guess to see if it matches a letter contained in the word.
4. The tasks repeated are the guessing process. Every guess would consist of the same task but updated based on the previous guess' made. These would repeat until the user either runs out of guess' or they guess the entire word correctly.
5. The bad inputs would be if the user enters a non-string letter, such as an integer or a string consisting of more than one letter. These bad inputs only are for when the user is guessing. If the user also enters the mystery word, the bad inputs would be integers and letters. Overall, a number is a bad input.
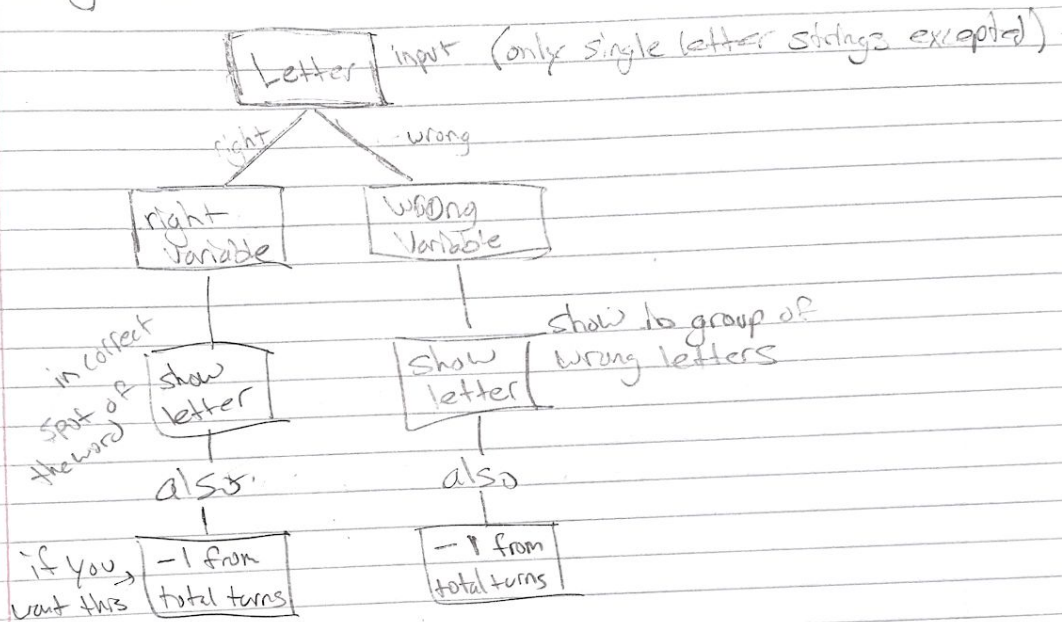
Step 3:
Mystery word- Bad: integers, letters   Good: a string longer then one letter and an actual word
        Max: infinite   Min: has to be at least a two letter word
Guess- Bad: integers, strings longer than one letter   Good: a letter
        Max: one letter in the english alphabet   Min: none

Design 1

guess function that is repeated

Letter — input (only single letter strings excepted)

right — wrong

right Variable | wrong Variable

in correct spot of the word | show letter — Show letter | show to group of wrong letters

also: | also

if you want this → -1 from total turns | -1 from total turns

How to store Word

Word — only word string inputs excepted

Variable — Word length = X

assign letter to a variable | assign letter to a variable

X number of times for each letter of word