

## Computer Science AP

### HangPerson Hints

For these hints, I will use the following class variables:

*mystery* the word the user is trying to guess

*guessed* a string containing the letters that the user has guessed

*showword* the string you will show the user (ex. co-put-r)

1. Assume the person enters a single, lower case letter in a textbox when they make their guess. Later on you can modify this to make sure that they are entering valid guesses.
2. How to make sure the user only enters a single letter?  
You know how to check the length of a string. So before you do anything with the user's guess, check to see if the length is 1. If not, just use the line `return;` to leave the method.
3. One way to keep track of the letters previously guessed is to create a String called *guessed*. *guessed* can start the game equal to "" (an empty string). Each time a single letter is guessed, add it on to this string with `+`. The string will continue to grow containing all the letters guessed. So if the user has guessed 'asdfn' then the guessed string will equal "asdfn".
4. How do you know if the users guess is good or bad?  
The user has just made a guess. Lets say that you store it in a variable called *guess*. You can use a String method to determine if the letter *guess* is somewhere in *mystery*.
5. How to check if the user has guessed this letter before?  
You are keeping track of the letters guessed. Go check to see if this letter has already being guessed before. If it has, tell the user and then leave this method by using `return;` .
6. How to form *showword*. This is the section that gives students the most trouble each year. It's a good puzzle to try to solve on your own. Before reading the algorithm on the next page I would strongly urge you to try to solve this on your own.

Ok, here's one possible way to form *showword*.  
For starters, let's assume some values for our variables.  
*mystery* = "computer science"  
*guessed* = "cmetsn"

set showword equal to nothing (showword = "")  
now loop through each letter of *mystery*. the first letter would be 'c', then 'o', ...  
for each letter, ask if this letter is present in the guessed letter string.  
if it is  
    add on the current letter to showword with + symbol  
if it isn't  
    add on a "-" to showword with the + symbol

By the time you finish looping through each letter of *mystery* you will have formed the entire showword. Give this method some thought. Given the sample variables above, the loop would make showword grow as follows:

start looping through *mystery*  
""  
check for c, add c or -  
check for o, add o or -  
check for m, add m or -  
check for p, add p or -  
check for u, add u or -  
check for t, add t or -  
check for e, add e or -  
check for r, add r or -  
check for space, add space or -  
check for s, add s or -  
check for c, add c or -  
...etc...  
showword will build up to be "c-m—te- sc-ence"