

Hangman

1. Problem Description

Based on the phenomenal success on previous projects, your angel investor has come back to your firm for yet another application. You are to develop a simple 'Hangman' game in which the computer selects a random word and the child repeatedly guesses letters until the word is uncovered. Sensitive to the tender age of our target audience, we will NOT draw a partially executed criminal but simply keep track of the number of guesses. There is no maximum number of guesses.

With each successive guess, show the word with the correct letters filled in, and also all the letters that have been guessed so far. If the child guess the same letter more than once, do not count that guess against them. The guesses should be converted to upper case.

Provide a set of at least ten words to guess from. The instructor's example uses the words "one" through "ten", however you are free to provide as many words as you would like.

Keep track of the number of guesses the child takes and display this number at the end. Your application should only play the game once per execution (i.e. no "Would you like to play another game?" is required).

2. Notes

- Turn in only your source files.

3. Required Main Class

Hangman

4. Required Input

A series of uppercase or lowercase letters

5. Required Output

Your output should look something like the following example. Notice the multiple, repeated guesses (h and e). It must include your name.

Hangman - E. Eckert

_ _ _ _ _ Used letters: {}

Enter a letter: a

_ _ _ _ _ Used letters: {A}

Enter a letter: b

_ _ _ _ _ Used letters: {AB}

Enter a letter: c

_ _ _ _ _ Used letters: {ABC}

Enter a letter: e

_ _ _ E E Used letters: {ABCE}

Enter a letter: t

T _ _ E E Used letters: {ABCET}

Enter a letter: h

T H _ E E Used letters: {ABCETH}

Enter a letter: h

Enter a letter: e

Enter a letter: r

T H R E E Used letters: {ABCETHR}

You guessed it in 7 tries.