# CMPS4143 Contemporary Programming Languages

**Program Assignment 1: 75 points**

**DUE: Wednesday, September 16, 2015**

**Purpose:** To develop and write your first Java program; to use interactive input and output in Java; to use control (conditional and iterative) statements; and to declare, define, and use a class.

**Problem:** Consider a guessing game in which a player tries to guess a hidden word. The hidden word contains only capital letters and has a length known to the player. A guess contains only capital letters and has the same length as the hidden word.

After a guess is made, the player is given a hint that is based on a comparison between the hidden word and the guess. Each position in the hint contains a character that corresponds to the letter in the same position in the guess. The following rules determine the characters that appear in the hint.

|  |  |
| --- | --- |
| **If the letter in the guess is…** | **the corresponding character in the hint is** |
| also in the same position in the hidden word, | The matching letter |
| also in the hidden word, but in a different position, | “+” |
| not in the hidden word, | “-“ |

The HiddenWord class will be used to represent the hidden word in the game. The hidden word is passed to the constructor. The class contains a method, getHint, that takes a guess and produces a hint.

For example, suppose the variable object puzzle is declared as follows:

HiddenWord puzzle = new HiddenWord (“HARPS”);

The following table shows several guesses and the hints that would be produced.

|  |  |
| --- | --- |
| **Call to getHint** | **String returned** |
| puzzle.getHint(“AAAAA”) | “+A+++” |
| puzzle.getHint(“HELLO”) | “H----“ |
| puzzle.getHint(“HEART”) | “h-++-“ |
| puzzle.getHint(“HARMS”) | “HAR-S” |
| puzzle.getHint(“HARPS”) | “HARPS” |

**Method:** Write the complete HiddenWord class, including any necessary instance variables, its constructor, and the method, getHint, described above. In getHint, you may assume that the length of the guess is the same as the length of the hidden word, however, your program may not assume it.

Write a program to

* Initialize the hidden word – ideally we’d read this from a file, but oh well…we’ll get it from the keyboard and assume the guesser closes their eyes while another user enters the hidden word.
* Convert the hidden word to capitals
* Get the length of the hidden word and show it to the user
* Repeatedly, ask the user for a guess
  + Check its length
  + Get a guess
  + Convert it to caps
  + Count the guess
  + Call getHint
  + Print the hint
* Print the hidden word, the number of guesses, and the message.

**Input:**  Input is from the keyboard. The user should be prompted to enter a string (in all capitals) for the hidden word. (However, the user may not follow directions. So, convert the string to capitals.) The user who guesses should be prompted to enter strings (in all capitals) for guesses. (However, this user may not follow directions. So, convert the string to capitals.)

**Output:** Output is to the monitor. Output should be clearly labeled and formatted. The program should print hints until the user guesses the word or reaches 10 guesses. It should print the number of guesses tried. It should print a message commenting on the number of guesses the user tried. (For example, if the user guessed it correctly in 1 guess – print “You must have ESP.)

**Turn in:**

1. Printout of the *source* program.
2. Set of three printouts of outputs. The first printout uses the tests above. Make up two others. MAKE SURE YOUR PROGRAM RUNS ON OTHER INPUTS!
3. Disk with source and class files on it – I will run it.
4. Put it in an approx. 8.5x11 envelope

The final program will be graded based on understandability of prompts, clarity of output, comments, indentation style, use of concepts in the purpose statement and, of course, correctness.