



## APCS Practice Problem: Hidden Word Puzzle

Consider a word 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 attempt 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 `HiddenPhrase` 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, `getFeedBack`, that takes a guess and produces a hint. For example, suppose the variable `puzzle` is declared as follows.

```
HiddenPhrase puzzle = new HiddenPhrase("HARPS");
```

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

Call to `getFeedback`                      String returned

<code>puzzle.getFeedback("AAAAA")</code>	<code>"+A+++"</code>
<code>puzzle.getFeedback("HELLO")</code>	<code>"H*****"</code>
<code>puzzle.getFeedback("HEART")</code>	<code>"H*++*"</code>
<code>puzzle.getFeedback("HARMS")</code>	<code>"HAR*S"</code>
<code>puzzle.getFeedback("HARPS")</code>	<code>"HARPS"</code>

Write the complete `HiddenPhrase` class, including any necessary instance variables, its constructor, and the method, `getFeedback`, described above. You may assume that the length of the guess is the same as the length of the hidden word.

```
public class HiddenPhrase {
    private String secretPhrase;

    public HiddenPhrase(String word){
        secretPhrase = word;
    }

    public String getFeedback(String guess){
        String temp=""; //our return value

        //Loop through all letters of guess assuming its same size
        for(int i =0; i<guess.length; i++){
            //3 cases, matches, in word wrong spot or not in word

            if(guess.substring(i,i+1).equals(secretPhrase.
                substring(i,i+1)){
                //Matches
                temp += guess.substring(i,i+1);
            }
            else if (secretPhrase.indexOf(guess.substring(i,i+1))!=-1){
                temp += "+";
            }
            else {
                temp += "*";
            }
        }
        return temp;
    }
}
```

