Hangman Problem (object_oriented)

Create a hangman program.

```
Sample output from your program should look like the following:

Current Status for userInputs=
 _ _ _ _ _ _
Enter next letter: a

Current Status for userInputs=a
 _ _ _ _ _ _
Enter next letter: i

Current Status for userInputs=ai
 _ i _ _ _ _
Enter next letter: e

Current Status for userInputs=aie
 _ i _ _ _ e
Enter next letter: s

Current Status for userInputs=aies
 s i _ _ _ e
Enter next letter: g

Current Status for userInputs=aiesg
 s i _ _ _ e
Enter next letter: m

Current Status for userInputs=aiesgm
 s i m _ _ e
Enter next letter: p

Current Status for userInputs=aiesgmp
 s i m p _ e
Enter next letter: l

Current Status for userInputs=aiesgmpl
 s i m p l e

Congratulations: you guessed the word!!
Do you want to play object oriented Hangman again? (y or n): y

Current Status for userInputs=
 _ _ _ _ _ _
Enter next letter: a

Current Status for userInputs=a
 _ _ _ _ _ _
Enter next letter: e

Current Status for userInputs=ae
 _ _ _ _ e _
```

```
Enter next letter: i

Current Status for userInputs=aei
 _ _ _ _ e _
Enter next letter: o

Current Status for userInputs=aeio
 _ o _ _ e _
Enter next letter: m

Current Status for userInputs=aeiom
 m o _ _ e _
Enter next letter: u

Current Status for userInputs=aeiomu
 m o _ _ e _
Enter next letter: n

Current Status for userInputs=aeiomun
 m o n _ e _
Enter next letter: k

Current Status for userInputs=aeiomunk
 m o n k e _
Enter next letter: y

Current Status for userInputs=aeiomunky
 m o n k e y

Congratulations: you guessed the word!!
Do you want to play object oriented Hangman again? (y or n): n
Bye
```

****
****

The following template gives you a starting point:

```
package object_oriented;

import java.util.*;

public class Hangman
{
    Scanner keyboard = new Scanner(System.in);
    Random rand = new Random();

    // The following routine will determine if the character c
    // is inside the String str.  A true is returned if it is inside.
    // It is very useful to call the isIn routine inside of printCurrStatus
...
    // See the comments in the Hint for printCurrStatus.

    boolean isIn(char c, String str)
    {
        //********** Fill in Details
    }

 // ******   printCurrStatus

     // If userInputs contains "ard" and strToGuess contains "aardvark" then
     // the following routine prints out an output that looks something like:
     //
     // Current Status for userInpts=ard
     // a a r d _ a r _

     // This routine returns true if all letters were guessed, otherwise
false is returned.

 // HINT:  It is useful to have a for loop that goes through each of the
characters in
//         strToGuess.    Call isIn for each character (note the second
parameter would
//         be userInputs).   If isIn returns true, just print out the
character, if isIn
//         returns false, then print out  '_'.
//         Additionally, you can have a variable like:
//         boolean success = true;
//         Whenever you output at least one '_', you can set success = false.
//         Your code can just return the variable "success" and it will
return true if
//         the user has picked all of the letters.



    boolean printCurrStatus(String strToGuess, String userInputs)
    {
        //********** Fill in Details
    }

    // The following routine will return a random String from the list of
words:
    // elephant, tiger, monkey, baboon, barbeque, giraffe,  simple, zebra,
    // porcupine, aardvark

    String getNextWordToGuess()
    {
        final int num_words=10; // change this if you have a different number
of words
        int num = rand.nextInt(num_words);
        // Another way to accomplish the same thing:
        // int num = (int)(num_words* Math.random());

        //********** Fill in Details
    }

    // The following routine plays the hangman game. It calls
getNextWordToGuess to
    // get the word that should be guessed.  It then has a loop which outputs
the
    // following prompt:
```

```
    // Enter next letter
    //
    // A String named userInputs stores all letters selected already.
    // Then the routine printCurrStatus is called to print out the current
status of
    // the guessed word.  If printCurrStatus returns true, we are done.

    void playGame()
    {
        //********** Fill in Details

    }

    // main will call playGame to play the hangman game.
    // Then main will issue the prompt:
    // Do you want to play again (y or n)
    // If the answer is "y", then call playGame again, otherwise exit

    public static void main(String[] args)
    {
        Hangman hangman = new Hangman();

        String response="";
        do
        {
            hangman.playGame();
            System.out.print("Do you want to play object oriented Hangman
again? (y or n): ");
            response = hangman.keyboard.next();
        } while (response.charAt(0) == 'y');

        System.out.println("Bye");
    }
}
```