```
'''
File: wordgame.py
Authors: put your last names here
Description:
    Implements a word guessing game similar to Hangman
'''

# Import statement: DO NOT delete these! DO NOT write code above this!
import random

# -----------------------------------
# Helper code
# (you don't need to understand this helper code)
# Import words for the game

WORDLIST_FILENAME = "words.txt"

def load_words():
    """
    Returns a list of valid words. Words are strings of lowercase letters.

    Depending on the size of the word list, this function may
    take a while to finish.
    """
    print "Loading word list from file..."
    # inFile: file
    inFile = open(WORDLIST_FILENAME, 'r', 0)
    # line: string
    line = inFile.readline()
    # wordlist: list of strings
    wordlist = line.split()
    print "  ", len(wordlist), "words loaded."
    print 'Type the function name for the version you want to play and press Enter
to play a game!'
    return wordlist

# load the dictionary of words and point to it with
# the words_dict variable so that it can be accessed from anywhere
# in the program
words_dict = load_words()


# Run get_word() within your program to generate a random secret word
# by using a line like this within your program:
# secret_word = get_word()

def get_word():
    """
    Returns a random word from the word list
    """
    word=words_dict[random.randrange(0,len(words_dict))]
    return word

# end of helper code
# -----------------------------------


# CONSTANTS
MAX_GUESSES = 6


def print_guessed(secret_word, letters_guessed):
    '''
```

```
    Receives two string arguments called secret_word and letters_guessed
    Returns a string which is the string secret_word with a dash used
    to replace each character that has not yet been guessed by the player
    '''

    word_so_far = ''

    #for each character in secret_word
    for c in secret_word:

        #if character is in letters_guessed, adds it to word_so_far
        #if not, adds a dash to word_so_far
        if c in letters_guessed:
            word_so_far = word_so_far+c
        else:
            word_so_far = word_so_far + '-'

    return word_so_far
    # word_so_far puts together all of the previous correct guesses
    # and the blanks that represent the letters not yet guessed.


def play_wordgame_v1():
    """
    Description:
        Welcomes player to game
        Chooses random word
        Prompts player to guess letter
        Receives guessed letter
        Determines whether guess is correct or incorrect
        When certain number of wrong guesses made, you lose
        When word is guessed correctly before losing all guesses, you win
    """
    # play the game

    #pick a random word from word list
    secret_word = get_word()
    guesses_made = 0

    letters_guessed = ''
    # at the begining, no letters have been guessed

    print "Welcome to the word guessing game!"
    print "I am thinking of a word that is "+str(len(secret_word))+" letters long."

    while guesses_made < MAX_GUESSES:
        #calculates and prints remaining number of guesses
        guesses_left = MAX_GUESSES - guesses_made
        print "You have "+str(guesses_left)+" guesses left."

        #asks user to enter a letter
        letter = raw_input("Please enter a letter: ").lower()

        #This conditional statement ensures that the user enters a
        #single letter
        if len(letter)>1:
            letter = raw_input("Please enter a single letter: ").lower()
        elif letter.isalpha():
            pass
        elif letter.isdigit():
            letter = raw_input("Please enter a LETTER: ").lower()
        else:
            letter = raw_input("Please enter a LETTER: ").lower()
```

```
        # this checks if the letter has already been guessed,
        #and if it has, ensures that a guess is not taken away
        if letter in letters_guessed:
            print "Oops! You have already guessed this letter."

        #if it has not been guessed and is a new letter, then it is
        #added to the string of letters guessed so far
        else:
            letters_guessed = letters_guessed + letter

        # this shows the word so far
            word_so_far = print_guessed(secret_word, letters_guessed)

            #let user know if letter is in secret word
            if letter in secret_word :
                print "Good guess: "+word_so_far

                #exit loop to win game
                win = True
                #condition for win is False - if a character from
                #secret_word is not in the letters guessed so far
                for i in range(len(secret_word)):
                    if secret_word[i] not in letters_guessed:
                        win = False
                        break
                if win:
                    print "You win!"
                    break

            #let user know if letter is not in secret word
            #number of guesses made so far is incremented by 1
            elif letter not in secret_word:
                print "Oops! That letter is not in my word: "+word_so_far
                # number of wrong guesses increases by 1
                guesses_made = guesses_made + 1


    print "Game over! The correct word is "+secret_word+"."
    #allows you to play the game again with another random word
    print 'Type play_wordgame_v1() and press Enter to play a game!'

    return None


def play_wordgame_v2():
    """
    Launches game for user, determines victory or defeat, prints game progress
    """
    secret_word = get_word()

    guesses_made = 0

    letters_guessed = ""
    word_so_far = ""


    #Welcomes messages

    print "See if you can guess this " + str(len(secret_word))\
          + " letter word!"
    print "You may make up to 6 incorrect guesses"
```

```
while True:
    #Winning condition
    if secret_word == word_so_far:
        #print word_so_far
        print "Congratulations! You win!"
        break

    #If user still has guesses
    elif guesses_made < MAX_GUESSES:
        print "You have " + str(MAX_GUESSES - guesses_made)\
        + " incorrect guesses remaining"
        last_guess = raw_input("Please enter a letter: ")
        last_guess = last_guess.lower()

    #This while loop is what keeps the game going after the first round
    # First section makes sure the user guesses a new letter
        while True:
            if last_guess in letters_guessed:
                print "You already guessed that letter!"
                break
            #Adds new letter to set of guessed letters
            else:
                letters_guessed = letters_guessed + last_guess
                word_so_far = print_guessed(secret_word, letters_guessed)

            #If corrrect guess made
                if last_guess in secret_word:
                    print "Good guess!"
                    print "Here is what you have so far: "
                    print word_so_far
                    print "You have guessed these letters: "
                    print letters_guessed
                    break
                else:
            #If incorrect guess made
                    if last_guess not in secret_word:
                        guesses_made = guesses_made + 1
                        print "You have made " + str(guesses_made)\
                          + " incorrect guesses"
                        print "So far, you have correctly guessed: "
                        print word_so_far
                        print "You have guessed these letters: "
                        print letters_guessed
                        break
                    else:
                        break

    #Losing condition
    else:
        print "Sorry, you lose!"
        print "The word you were looking for was: "
        print secret_word
        break

return None
```