

TDD: Top Down Design*(find and work with a partner)*

Today we'll be starting Top Down Design (TDD), a way of designing the structure of a program so that implementation and testing work more smoothly. TDD also helps create code that is easier to read, debug, and modify. TDD is usually followed by "bottom-up" implementation of the functions outside of main. The following code is an outline for a word guesser game.

1. First fill in the blank sections in the code to complete main.
2. On the back of this sheet or a separate sheet, list all the functions that need to still be implemented.
3. For each function, list the parameters, describe in words what the function should do, and specify its return type.

```
def main():

    # define a hidden word and a list of spaces for the user of the same length
    hidden_word = "myword"
    user_lst = len(hidden_word)*["_"]
    bad_letters = []
    guessed = False

    # keep asking for letters until the user has guessed the word or reached 6 bad letters

    while _____:

        # print the current progress and get the user's guess
        display_round(user_lst, bad_letters)
        letter = get_guess(user_lst, bad_letters)

        # if the letter was in the hidden word, update the current progress
        if letter in hidden_word:
            process_letter(letter, hidden_word, user_lst)

        # otherwise, add the letter to bad_letters
        else:
            _____

        # if the user has completed the hidden word, what should happen?
        if correct(user_lst, hidden_word):
            _____

    # after the game is over, display the end result and print messages to user
    display_round(user_lst, bad_letters)
    if guessed:
        print("_____")
    else:
        print("_____")
```

- Function 1:

- Function 2:

- Function 3:

- Function 4: