MITx: 6.00x Introduction to Computer Science and Programming

Ankit_Aggarwal (/dashboard) ▾

Courseware (/courses/MITx/6.00x/2012_Fall/courseware) Course Info (/courses/MITx/6.00x/2012_Fall/info)

Textbook (/courses/MITx/6.00x/2012_Fall/book/0/) Discussion (/courses/MITx/6.00x/2012_Fall/discussion/forum)

Wiki (/courses/MITx/6.00x/2012_Fall/course_wiki) Progress (/courses/MITx/6.00x/2012_Fall/progress)

TTI STA T D

1. **Download and sa e Problem Set (/static/content mit 600x 2012_Fall/files/templates/ProblemSet .d b2a1c c 6. ip) a ip file of all the skeleton code ou ll be filling in. e sure to sa e** *all the files in this i folder* ps4a.py,ps4b.py,test_ps4a.py and words.txt - in the **same folder. We recommend creating a folder in our Documents folder called 6.00 and inside the 6.00 folder creating a separate folder for each problem set. If ou don t follow this instruction ou ma end up with issues because the files for this problem set depend on one another.**

2. **un the file** ps4a.py, without making any modifications to it, in order to ensure that everything is set up correctly (this means, open the file in IDLE, and use the Run command to load the file into the interpreter). The code we have given you loads a list of valid words from a file and then calls the playGame function. You will implement the functions it needs in order to work. If everything is okay, after a small delay, you should see the following printed out

```
Loading word list from file...
      83667 words loaded.
playGame not yet implemented.
```

   If you see an IOError instead (e.g., *No such file or directory*, *you should change the value of the* WORDLIST_FILENAME constant (defined near the top of the file) to the **complete pathname for the file** words.txt (This will vary based on where you saved the file).

   ost in the forum if you are having further issues with this.

3. The file ps4a.py has a number of already implemented functions you can use while writing up your solution. You can ignore the code between the following comments, though you should read and understand how to use each helper function by reading the docstrings

```
# -----------------------------------
# Helper code
# You don't need to understand this helper code,
# but you will have to know how to use the functions
# (so be sure to read the docstrings!)
    .
    .
    .
# (end of helper code)
# -----------------------------------
```

4. This problem set is structured so that you will write a number of modular functions and then glue them together to form the complete word playing game. Instead of waiting until the entire game is *ready, you should test each function you rite, individually, efore moving on. his a roach is no n as unit testing, and it ill hel you de ug your code.*

test_ps4a.py to check your work.

If your code passes the unit tests you will see a SUCCESS message otherwise you will see a FAILURE message. These tests aren't exhaustive. You will want to test your code in other ways too.

Try running test_ps4a.py now (before you modify the ps4a.py skeleton). You should see that all the tests fail, because nothing has been implemented yet.

These are the provided test functions

test_getWordScore()
    estt e getWordScore() implementation.

test_updateHand()
    estt e updateHand() implementation.

test_isValidWord()
    estt e isValidWord() implementation.

ow s ss o

New Post