
CMSC 201 Fall 2016

Homework 8 – House Sorting

Assignment: Homework 8 – House Sorting

Due Date: Wednesday, November 23rd, 2016 by 8:59:59 PM

Value: 40 points

Collaboration: For Homework 8, collaboration is allowed. Make sure to consult the syllabus about the details of what is and is not allowed when collaborating. You may not work with any students who are not taking CMSC 201 this semester. If you work with someone, remember to note their name, email address, and how you collaborated at the top of your file.

If you did not work with anyone else on this assignment, your collaboration statement should state that

I did not collaborate with anyone on this assignment.

If you did work with someone else (or multiple other people), your collaboration statement should state something *similar* to the following:
I collaborated with Fox Mulder (fmulder1@umbc.edu); I helped him understand the loop.

Make sure that you have a complete file header comment at the top of your file, and that all of the information is correctly filled out.

```
# File:      FILENAME.py
# Author:    YOUR NAME
# Date:      THE DATE
# Section:   YOUR DISCUSSION SECTION NUMBER
# E-mail:    YOUR_EMAIL@umbc.edu
# Description:
#   DESCRIPTION OF WHAT THE PROGRAM DOES
# Collaboration:
#   COLLABORATION STATEMENT GOES HERE
```

Homework 8 is designed to help you practice using dictionaries, file input, and Python's `random` library. You will also get more practice with creating and calling functions.

Remember to enable Python 3 before running and testing your code:

```
scl enable python33 bash
```

Instructions

In this homework, you will be doing a series of exercises designed to help you practice using dictionaries, file input, and Python's `random` library. For this assignment, you may assume that all the input you get will be of the correct type (e.g., if you ask the user for a whole number, they will give you an integer).

For this assignment, you'll need to follow the class coding standards, a set of rules designed to make your code clear and readable. The class coding standards are on the website, linked at the top of the "Assignments" page. You can also access them directly via this URL (<http://goo.gl/yEoGfC>).

At the end, your Homework 8 file must run without any errors.

Details

Homework 8 is broken up into four separate parts. **Make sure to complete all four parts.**

Although Homework 8 is broken up into separate parts, you will be turning in a **single file** at the end, with all of your functions contained inside it. The single file should be called `hw8.py`.

NOTE: Your filename for this homework must match the given one exactly.
And remember, filenames are case sensitive!

Parts

Each part is worth the indicated number of points. Following the coding standards, having complete file headers, and having the correctly named file is worth 4 points.

For Homework 8, you will be creating a **single file**, called `hw8.py`, that contains **at least** these four separate functions:

- `printOneHouse()`
- `printAllHouses()`
- `houseSort()`
- `main()`

You are welcome to create additional functions as you see fit – we expect that most students will want to use at least two or three additional functions.

Each function is described in detail below. **Adhere to the coding standards by including function header comments for all of your functions** (other than `main`). Follow the instructions and example provided in the coding standards document when creating your function header comments. Failing to include function header comments will lose you points.

You are highly encouraged to test each function fully before moving on to the next. It is recommended **not** to complete the looping functionality of `main()` until all of your other functions work correctly.

Using a “temporary” `main()` to interact with each function as you develop and test it may prove useful.

Pay close attention to the sample output provided when testing your own code.

Homework 8 is a simulation of the “Sorting Hat” from the Harry Potter books. It reads in the current house members from a file, storing them in a dictionary (**the key for each entry is the house name, and the value associated with it is a list of the house members**).

The program allows the user to see who is in each house, either by printing all of the houses, or by printing a single house at a time. It also allows a user to “sort” themselves into a house. Much like in the books, the person being sorted can make requests of the “Sorting Hat” program, but the end result is not entirely up to them. (In our program, the “Sorting Hat” uses a random number generator to help it decide what house to put someone in.)

Part1 - `printOneHouse()` (Worth 5 points)

The `printOneHouse()` function prints out the name of the house and all of the members for a single house in the house dictionary. If the house does not exist in the dictionary, it should print out an error message (see the sample output for an example), but should not crash or exit.

The input and output for this function are left up to you – think carefully about what information it needs to give and receive!

Part2 - `printAllHouses()` (Worth 4 points)

The `printAllHouses()` function prints out the name of the house, and all of the members, for each of the houses contained in the house dictionary. The input and output for this function are left up to you – think carefully about what information it needs to give and receive!

We recommend writing `printOneHouse()` first, and then calling that function inside of the `printAllHouses()` function.

Part3 – houseSort ()

(Worth 12 points)

The `houseSort ()` function is the most complex function, and randomly sorts a person into one of the available houses. The user is asked for their house preference, and the function gives that house a far greater chance of being picked.

The input and output for this function are left up to you – think carefully about what information it needs to give and receive!

Specifically, the preferred house has a chance of being picked equal to

$$\frac{n + 1}{2 * n}$$

when n is the number of houses available as an option.

For example, if there are 4 houses, the chance of picking the preferred house is 5 out of 8. The preferred house has the given 1 out of 4 chance, but also an additional 4 chances to be picked.

If there are 5 houses, the chance is 6 out of 10. (5 additional chances)

If there are 2 houses, the chance is 3 out of 4. (2 additional chances)

If there are 10 houses, the chance is 11 out of 20. (10 additional chances)

You may implement this using any of the functions we covered from Python's `random` library, but make sure that you are giving the correct amount of “preference weight” to the user's preferred house.

(Note that since we are using random numbers, it is possible to be “unlucky” multiple times in a row. This part will be graded based on your code, not on the exact selections made when we run your program.)

Make sure to comment this function well, as there are many different ways of taking into account the person's preference and implementing the selection.

Part4 - main()

(Worth 6 points)

The `main()` function should handle all of the other functions (including any others you may write) and the way the program runs. The program should:

- Ask the user for a file to read in the information from
 - **See the sample input files for the format used**
- Present the user with a menu
- As long as the user does not choose to quit:
 - Ask the user for their menu choice
 - If they choose to print a single house
 - Ask the user for the house name to print
 - Print out the house name and its members, or an error message if the house doesn't exist
 - If they choose to print all houses
 - Print out the house names and their members
 - If they choose to sort a new person into a house
 - Ask for the person's name
 - Ask for the person's house preference
 - **You can assume this is a valid, existing house name**
 - Print out the result of the sorting
 - Reprint the menu and reprompt the user for their menu choice
 - ❖ Continue until the user has chosen to quit the program

Part5 – Everything Else

(Worth 9 points)

The rest of your program is also worth points, including

- Printing a menu
- Input validation on the menu choice
- Reading in the data from a file
- Creating and using a dictionary

is also worth points.

Whether you decide to use function for each of these tasks, or you include them within `main()` or another function is up to you.

See the following pages for sample output, with the user input in blue.

We have also highlighted the results in orange, to make it easier to pick them out from all of the menu choices being printed out.

(Yours does not have to match this exactly, but it should be similar.)

NOTE: Since the houses and their members are stored in a dictionary, and since random numbers are used to sort people into houses, your output and results will almost certainly differ from those shown below.

You can download [sample input files](#) from Dr. Gibson's public directory:

```
cp /afs/umbc.edu/users/k/k/k38/pub/cs201/HP.txt .  
cp /afs/umbc.edu/users/k/k/k38/pub/cs201/short_HP.txt .  
cp /afs/umbc.edu/users/k/k/k38/pub/cs201/GoT.txt .
```

All of the files we use for testing your program will follow this formatting.

```

bash-4.1$ python hw8.py
Please enter filename to load from: short_HP.txt

Please make a choice from the menu:
1 - Print a single house
2 - Print all the houses
3 - Sort a new person into a house
4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 1
What house's members would you like to print? AwesomeSauce
There is no house by the name of AwesomeSauce

-----

Please make a choice from the menu:
1 - Print a single house
2 - Print all the houses
3 - Sort a new person into a house
4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 1
What house's members would you like to print? Huff
The members of the House of Huff are:
    Cedric Diggory

-----

Please make a choice from the menu:
1 - Print a single house
2 - Print all the houses
3 - Sort a new person into a house
4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 3
What is the person's name? Liz
What house do they prefer? Huff
Liz was sorted into house Huff

-----

```

(Additional sample output on the next page.)

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 1

What house's members would you like to print? Huff

The members of the House of Huff are:

Cedric Diggory

Liz

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 2

The members of the House of Gryff are:

Hermione Granger

Harry Potter

Ginny Weasley

Ron Weasley

The members of the House of Raven are:

Cho Chang

Luna Lovegood

Padma Patil

Myrtle Warren

The members of the House of Slyth are:

Draco Malfoy

Andromeda Tonks

The members of the House of Huff are:

Cedric Diggory

Liz

(Additional sample output on the next page.)

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 3

What is the person's name? Patrick

What house do they prefer? Raven

Patrick was sorted into house Slyth

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 3

What is the person's name? Maria

What house do they prefer? Raven

Maria was sorted into house Raven

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 3

What is the person's name? Jacob

What house do they prefer? Slyth

Jacob was sorted into house Slyth

(Additional sample output on the next page.)

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 2

The members of the House of Gryff are:

Hermione Granger
Harry Potter
Ginny Weasley
Ron Weasley

The members of the House of Raven are:

Cho Chang
Luna Lovegood
Padma Patil
Myrtle Warren
Maria

The members of the House of Slyth are:

Draco Malfoy
Andromeda Tonks
Patrick
Jacob

The members of the House of Huff are:

Cedric Diggory
Liz

Please make a choice from the menu:

- 1 - Print a single house
- 2 - Print all the houses
- 3 - Sort a new person into a house
- 4 - Exit the program

Please enter a number between 1 and 4 (inclusive): 4

Thank you for using the Great Houses Program

Submitting

Once your `hw8.py` file is complete, it is time to turn it in with the `submit` command. (You may turn in the file multiple times as you complete each function. To do so, simply submit `hw8.py` each time you complete a part of the homework. Each new `submit` will overwrite the old file.)

You must be logged into your GL account, and you must be in the same directory as your Homework 8 python file. To double-check this, you can type `ls`.

```
linux1[3]% ls
hw8.py
linux1[4]% █
```

To submit your Homework 8 python files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW8`. Type in (all on one line) `submit cs201 HW8 hw8.py` and press enter.

```
linux1[4]% submit cs201 HW8 hw8.py
Submitting hw8.py...OK
linux1[5]% █
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can check that your homework was submitted by following the directions in Homework 0. Double-check that you submitted your homework correctly, since **an empty file will result in a grade of zero for this assignment.**