
CMSC 201 Fall 2015

Homework 7 – Functions

Assignment: Homework 7 – Functions

Due Date: Thursday, October 29th, 2015 by 8:59:59 PM

Value: 4% of final grade

Homework 7 is designed to help you practice using functions and incremental programming. One of the goals of using functions is to break up larger tasks into smaller, more manageable tasks. Similarly, when we are programming an entire application, we try to break it down and program each part incrementally (or even function by function). More importantly, you will be solving problems using algorithms you create and code yourself.

Remember to enable Python 3 before you run your programs:

```
/usr/bin/scl enable python33 bash
```

Instructions

We will be completing this assignment in ONE python file. It should be named hw7.py. 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 Blackboard under “Course Documents” in a file titled “CMSC 201 - Python Coding Standards.”

You will **lose major points** if you do not following the 201 coding standards.

A very important piece of following the coding standards is writing a complete **file header comment block**. Make sure that each file has a comment block at the top (see the coding standards document for an example).

NOTE: You must use `main()` in your file.

Details

Homework 7 is one complete application that will be a calculator. **When you are finished, you will be submitting one file with many functions.**

NOTE: Your filename for this homework must match the given one exactly.

And remember, filenames are case sensitive.

hw7.py

You will be writing a homework using functions to get a list of integers from a user and then find statistics of the given list of integers. You are not allowed to use any built-in functions for finding the calculations in this assignment (although you can use the `sort()` function for median).

The functions required for this assignment will be:

`main()` – Primary driver for the program.

`printMenu()` – Prints a menu of options to the screen.

`getList()` – Gets list of integers from user using a while loop and returns the list. Stops asking for more input if the user enters a 'q'.

`getMean(userList)` – Calculates the average value of the set using a for loop and prints value to screen. Do not use any third-party functions to get the mean.

`getMedian(userList)` – Finds the median value in the set using a for loop and then prints value to screen. You can use the `sort()` function to sort the list to get the median. Do not use any built-in/third-party functions to get the median.

`getMin(userList)` – Finds the lowest value in the set using a for loop and then prints value to screen. Do not use any built-in/third-party functions to get the minimum.

`getMax(userList)` – Finds the highest value in the set using a for loop and then prints value to screen. Do not use any built-in/third-party functions to get the maximum.

`emptyList(userList)` – Empties the current list and calls `getList()` again to repopulate the list.

Hint 1: Getting Started

We understand that this is a significantly more complex problem than we have done in homework previously. Break up the problem into individual steps and code each function separately. This incremental programming will make it easier to debug!

Hint 2: Scope

Remember that scope is important. Once you exit from a function, the variables in that function are removed from memory. Make sure that your list of integers gets passed to and from each function!

Hint 3: Median

The median of a set of data is the middlemost number in the set. The median is also the number that is halfway into the set. To find the median, the data should be arranged in order from least to greatest. If there is an even number of items in the data set, then the median is found by taking the mean (average) of the two middlemost numbers. As you need a sorted dataset to find the median, your hint is as follows: to sort a list, use the built-in function `listName.sort()`

Hint 3: Empty

There are a bunch of easy ways to clear a list. Generally, the easiest way to empty a list is the same way that we initialize an empty list.

Here is the sample output for hw7.py, with the user input in blue. Your output does not need to be identical, but should be similar.

```

bash-4.1$ python hw7.py
Welcome to the List Statistics Calculator
Enter a list of integers or q to end the list!
Enter an integer: 10
Enter an integer: 12
Enter an integer: 13
Enter an integer: 8
Enter an integer: 3
Enter an integer: 5
Enter an integer: 4
Enter an integer: q
Please choose the statistic you would like to calculate!
1. Min
2. Max
3. Mean
4. Median
5. Clear List
Please enter your choice, or 0 to exit: 1
The minimum value is: 3
Please choose the statistic you would like to calculate!
1. Min
2. Max
3. Mean
4. Median
5. Clear List
Please enter your choice, or 0 to exit: 2
The maximum value is: 13
Please choose the statistic you would like to calculate!
1. Min
2. Max
3. Mean
4. Median
5. Clear List
Please enter your choice, or 0 to exit: 3
The mean value is: 7.857142857142857
Please choose the statistic you would like to calculate!
1. Min
2. Max
3. Mean
4. Median
5. Clear List
Please enter your choice, or 0 to exit: 0
Thank you for using our calculator
bash-4.1$

```

Here is additional sample output for hw7.py:

```
bash-4.1$ python hw7.py
Welcome to the List Statistics Calculator
Enter a list of integers or q to end the list!
Enter an integer: 10
Enter an integer: 30
Enter an integer: 20
Enter an integer: 40
Enter an integer: q
Please choose the statistic you would like to calculate!
1. Min
2. Max
3. Mean
4. Median
5. Clear List
Please enter your choice, or 0 to exit: 4
25.0
Please choose the statistic you would like to calculate!
1. Min
2. Max
3. Mean
4. Median
5. Clear List
Please enter your choice, or 0 to exit: 0
Thank you for using our calculator
bash-4.1$
```

Additional Hints: This would be a good time to use **incremental programming!** Incremental development is when you are only working on a small piece of the code at a time, and testing that the piece of code works before moving on to the next piece. This makes it a lot easier to fix any mistakes.

You can create the function definitions and leave them empty to incrementally test them.

```
def getMin(userList):
    return
```

Start with the stuff you already know how to do!

Submitting

Once your Homework 7 is complete, it is time to turn it in with the `submit` command.

Don't forget to complete the header block comment for your file! Make sure that you updated the header block's file name and description for the file.

You must be logged into your GL account, and you must be in the same directory as the Homework 7 files. To double check this, you can type `ls`.

```
linux1[3]% ls
hw7.py
linux1[4]% █
```

To submit your files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW7`. Type in (all on one line) `submit cs201 HW7 hw7.py` and press enter.

```
linux1[4]% submit cs201 HW7 hw7.py
Submitting hw7.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 **double-check that your homework files were submitted** by using the `submitls` command. Type in `submitls cs201 HW7` and hit enter.

And you're done!