

CS 1301 Homework 9

Homework – Make it Functional!

Due: Monday, November 30th before 11:55pm

THIS IS AN INDIVIDUAL ASSIGNMENT!

You should work individually on this assignment. You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. Students may collaborate with only fellow students currently taking CS 1301, the TAs, and the lecturer. You should not exchange code or write code for others. For individual assignments, each student must turn in a unique program. Your submission must not be substantially similar to another student's submission. Collaboration at a reasonable level will not result in substantially similar code.

Scored out of 75 points

Files to Submit:

hw9.py (make sure to complete all 3 parts!)

If you need help, we have several resources to help you successfully complete this assignment:

- The TA Helpdesk – Schedule posted on class website.
- Email the TAs
- Jay's office hours

Notes:

Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).

Do not wait until the last minute to do this assignment in case you run into problems.

If you find a significant error in the homework assignment, please let a TA know immediately.

Simple Functions

For this homework you will re-write some of the functions from homework 1 using functional programming. The functions that must be written are:

1. machToFPS
2. sqPyramidVolume
3. makeChange

Careful though! There are some changes since homework 1! (Look at the parameters)

Function Name: **machToFPS**

Parameters:

machList –a list of speeds in mach

Return Value:

None

Description:

For every value in the machList (you can assume all values will be integers):

1. Convert the value to feet per second using the **map** function.
2. Print out the old and new value in the format:

- “____ mach is equivalent to ____ feet per second”

*1 mach = 1116.4370079 feet / second

Example:

```
python>>> machToFPS([1,5,3])
```

```
1 mach is equivalent to 1116.4370079 feet per second
```

```
5 mach is equivalent to 5582.1850395 feet per second
```

```
3 mach is equivalent to 3349.3110237 feet per second
```

Function Name: **sqPyramidVolume**

Parameters:

baseHeightList - a list of tuples that contain bases and heights
[(base, height), (base, height), ...]

volumeList - a list of pyramid volumes (some correspond to given base and heights, some need to be eliminated)

Return Value:

correctList - the list of volumes that correspond to the given bases and heights.

Description:

You are given a list containing tuples of bases and heights of pyramids (see below for an example of such list.) For every base/height value:

1. Calculate the volume of a square pyramid with the given base and height using the **map** function; note the volume of a square pyramid is calculated using the formula below:

$$\text{Volume} = (\text{Base} \times \text{Base} \times \text{Height}) / 3$$

After you have calculated all the valid volumes, use the **filter** function to eliminate any volumes from the volumeList that cannot be calculated from the given bases/heights.

Example:

```
>> myBaseHeight = [(1,3),(3,4)]
>> myVolumes = [13,1,12,9,4,2]
>> sqPyramidVolume(myBaseHeight,myVolumes)
[1, 12]
```

Function Name: **makeChange**

***note this function is doing the reverse of what it did in homework 1.**

Parameters:

changeList-a list containing the amount of dollars, quarters, dime, nickels, and pennies are to be added (in that order)

Return Value:

totalValue -the total amount of change in cents.

Description:

Use the **reduce** function to add the values of the coins into one value (hint: it will be helpful to first get the cent value of what is given. i.e. [1,1,1,1,1] → [100, 25, 10, 5, 1])

You may assume the changeList will always have 5 valid integers.

Example:

```
>> makeChange([1,3,1,1,4])
>>194
```

Grading Rubric

machToFPS

- Uses map function
- Value calculated correctly
- Prints outputs correctly

20pts

5pts

10pts

5pts

sqPyramidVolume

- Calculates volume for every base/height correctly
- Uses map function
- Uses filter function to filter list
- Correct list returned

30pts

10pts

5pts

10pts

5pts

makeChange

- Calculates correct amount of change
- Uses reduce function
- Returns correct value

25pts

10pts

10pts

5pts