
CMSC 201 Fall 2015

Homework 4 – For Loops

Assignment: Homework 4 – For Loops
Due Date: Tuesday, October 6th, 2015 by 8:59:59 PM
Value: 4% of final grade

Homework 4 is designed to help you practice using `for` loops, branching selection structures, strings, and mathematical operators. 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 python3 bash`

Instructions

In this homework, we will be doing a series of exercises designed to make you practice using `for` loops, control statements like `if/else`, mathematical operators, and algorithmic thinking. Each one of these exercises should be in a **separate python file**. 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 each of your files.

Details

Homework 4 is broken up into five parts. **Make sure to complete all 5 parts.**

NOTE: Your filenames for this homework must match the given ones exactly.

And remember, filenames are case sensitive.

hw4_part1.py

For this part of the homework you will write code to draw a right triangle.

Your program should prompt the user for these inputs, **in exactly this order:**

1. The height of their triangle
2. The symbol the triangle will be made of

For these inputs, you can assume the following:

- The height will be a positive integer

You should then use the symbol they chose to print a right triangle of the height they indicated.

Here is some sample output, with the user input in blue.

```
bash-4.1$ python hw4_part1.py
Please enter the height of your triangle: 2
Please enter a single character for the symbol: &
&
&&

bash-4.1$ python hw4_part1.py
Please enter the height of your triangle: 5
Please enter a single character for the symbol: W
W
WW
WWW
WWWW
WWWWW
```

hw4_part2.py

For this part of the homework you will write code to draw an empty box.

Your program should prompt the user for these inputs, **in exactly this order**:

1. The width of their box
2. The height of their box
3. The symbol the box will be made of

For these inputs, you can assume the following:

- The height and width will be positive integers

Use the symbol to draw a box of the height and width chosen by the user.

The box should not be filled in with the symbol – it should only be outlined.

Please note that on your terminal a square box may not look exactly square.

(HINT: Use the concatenate and repetition operators we learned to print the right number of characters for each line.)

Here is some sample output, with the user input in blue.

```
bash-4.1$ python hw4_part2.py
Please enter the width of your box: 6
Please enter the height of your box: 3
Please enter a single character for the symbol: #
#####
#      #
#####

bash-4.1$ python hw4_part2.py
Please enter the width of your box: 1
Please enter the height of your box: 1
Please enter a single character for the symbol: M
M

bash-4.1$ python hw4_part2.py
Please enter the width of your box: 20
Please enter the height of your box: 4
Please enter a single character for the symbol: X
XXXXXXXXXXXXXXXXXXXXXXXXX
X                          X
X                          X
XXXXXXXXXXXXXXXXXXXXXXXXX
```

hw4_part3.py

Next, you are going to write code that determines whether a subject can be studied or not.

(WARNING: This part of the homework is the most challenging, so budget plenty of time and brain power. And read the instructions carefully!)

First, you must ask the user to enter ten different subjects, and store those subjects in a list. Once you have all ten subjects, use the following two rules to determine how each subject should be printed back out to the user.

1. If the subject ends in "ology" print out:

You can study SUBJECT

2. Otherwise, simply print the subject:

SUBJECT

See the sample output below for an example of correct program behavior.

For the input to this question, you can assume the following:

- The words entered will be in all lowercase
- The words entered will be at least 6 characters long

(HINT: You will need to use two `for` loops for this part of the homework. One loop to read in the subjects, and one loop to print them out. For an example, look at the `updatedGroceries.py` code under the Lecture 08 Notes, inside the "Code from Prof Gibson's Sections" folder.)

(HINT: You will want to use string slicing to check if the string ends with "ology." Review Lecture 07 (Strings and Lists) for details on how to use slicing.)

Here is some sample output for hw4_part3.py, with the user input in blue.
(Yours does not have to match this exactly, but it should be similar.)

```

bash-4.1$ python hw4_part3.py

Please enter a word: kumquats
Please enter a word: theology
Please enter a word: biology
Please enter a word: carnology
Please enter a word: english
Please enter a word: computer science
Please enter a word: theologies
Please enter a word: hauntology!
Please enter a word: dragonology
Please enter a word: twix???

kumquats
You can study theology
You can study biology
You can study carnology
english
computer science
theologies
hauntology!
You can study dragonology
twix???

```

hw4_part4.py

Write a program that prints the numbers from 1 to 100 (inclusive), one per line. However, there are three special cases where instead of printing the number, you print a message instead:

1. If the number you would print is **divisible by 3**, print the message:
Better three hours too soon than a minute too late.
2. If the number you would print is **divisible by 5**, print the message:
Where do you see yourself in five years?
3. If the number you would print is **divisible by 3 and 5**, instead print out:
In the future, everyone will be world-famous for 15 minutes.

Here is some sample output, up to the number 22.

```
bash-4.1$ python hw4_part4.py
1
2
Better three hours too soon than a minute too late.
4
Where do you see yourself in five years?
Better three hours too soon than a minute too late.
7
8
Better three hours too soon than a minute too late.
Where do you see yourself in five years?
11
Better three hours too soon than a minute too late.
13
14
In the future, everyone will be world-famous for 15 minutes.
16
17
Better three hours too soon than a minute too late.
19
Where do you see yourself in five years?
Better three hours too soon than a minute too late.
22
```

hw4_part5.py

Finally, you are going to create a simple program that counts the number of times a character appears in a string.

Your program should prompt the user for these inputs, **in exactly this order**:

1. The string to search through
2. The character to look for

IMPORTANT: Your program must not be case sensitive! It must count every instance of the character, whether it appears in upper or lower case.

Here is some sample output, with the user input in blue.

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

```
bash-4.1$ python hw4_part5.py
Please enter a string: I meant what I said and I said
what I meant.
Please enter a character: i
The character 'i' appears 6 times in the string:
    'I meant what I said and I said what I meant.'

bash-4.1$ python hw4_part5.py
Please enter a string: Speak up. You have to project! If
people can't hear you, it doesn't matter what you say.
Please enter a character: E
The character 'E' appears 8 times in the string:
    'Speak up. You have to project! If people can't hear
you, it doesn't matter what you say.'

bash-4.1$ python hw4_part5.py
Please enter a string: 3.1415926535897932384626433
Please enter a character: 3
The character '3' appears 6 times in the string:
    '3.1415926535897932384626433'
```

Submitting

Once all five parts of your Homework 4 are complete, it is time to turn them in with the `submit` command.

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

You must be logged into your GL account, and you must be in the same directory as the Homework 4 files. To double check this, you can type `ls`.

```
linux1[3]% ls
hw4_part1.py  hw4_part3.py  hw4_part5.py
hw4_part2.py  hw4_part4.py
linux1[4]% █
```

To submit your files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW4`. Type in (all on one line)
`submit cs201 HW4 hw4_part1.py hw4_part2.py hw4_part3.py
hw4_part4.py hw4_part5.py`
and press enter.

```
linux1[4]% submit cs201 HW4 hw4_part1.py hw4_part2.py
hw4_part3.py hw4_part4.py hw4_part5.py
Submitting hw4_part1.py...OK
Submitting hw4_part2.py...OK
Submitting hw4_part3.py...OK
Submitting hw4_part4.py...OK
Submitting hw4_part5.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 all five homework files were submitted by using the `submitls` command. Type in `submitls cs201 HW4` and hit enter.

And you're done!