
CMSC 201 Spring 2016

Homework 2 – Order of Operations

Assignment: Homework 2 – Order of Operations
Due Date: Monday, February 15th, 2016 by 8:59:59 PM
Value: 40 points

Homework 2 is designed to help you practice order of operations and modulo (or “mod”) in a Python environment. Remember to enable Python 3:

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

As a reminder, mod is the remainder left over when one number is divided by another. For example, $7 \% 3 = 1$. The value of mod will always be from zero – where there is no remainder (e.g., $10 \% 5 = 0$), to 1 less than the mod number (e.g., $19 \% 5 = 4$).

Instructions

For the first 8 questions below, there are three steps.

1. Using a comment in your python code, hypothesize what the correct answer is using order of operations.
2. Code the expression using a `print()` command.
3. Comment on why your answer was correct or incorrect.

For questions 9 and 10 below, you need to add parentheses to the equation so that it evaluates to the target number given. Make sure you test that your solution works using Python 3.

For questions 11 and 12 below, you are given some pseudocode and asked to implement the solution in Python. For this exercise, you will only need to use concepts we have discussed in class such as variables, expressions, casting to an integer, and `print()`.

Make sure that you **think about and hypothesize the answer before coding it** up and having Python calculate the answer. (You won’t have Python available on the midterm, after all.)

At the end, your Homework 2 files must run without any errors.

Example 1:

Evaluate the following expression:

example1 = (10 + 5) * 5 - 3

1. What do you think example1 equals?
2. Code the expression
3. Were you correct? Why or why not?

```
# Example 1:
# Expected output: 72
example1 = (10 + 5) * 5 - 3
print("Example 1 evaluates to:", example1)
# Actual output: 72
# Explanation: Parentheses first (15), then multiplication (75),
#              then subtraction (72)
```

Example 2:

Evaluate the following expression:

example2 = (5 + 5) * 5 - 3 % 3

```
# Example 2:
# Expected output: 49
example2 = (5 + 5) * 5 - 3 % 3
print("Example 2 evaluates to:", example2)
# Actual output: 50
# Explanation: Parentheses first (10), then multiplication (50)
#              and modulo (0), then subtraction (50)
#              My mistake was 3 % 3 = 0, and I thought it was 1.
```

Example 3:

Add parentheses so that the equation evaluates to the target number:

example3 = (8 * 3) / 6 * (1 + 2)

target3 = 12

```
# Example 3:
# Given equation: 8 * 3 / 6 * 1 + 2
# Solved equation: (8 * 3) / 6 * (1 + 2)
# Target number: 12
example3 = (8 * 3) / 6 * (1 + 2)
print("Example 3 evaluates to:", example3, "and should be", 12)
```

NOTE: Python may give you a decimal for your whole number (e.g., 12.0 instead of 12 for Example 3). We'll discuss the reason for this later.

Questions

Each question is worth 3 points. Following the directions is worth 4 points.

These questions should all be answered in a file called `hw2_part1.py`.

Question 1

`num1 = (7 + 4) * 5`

Question 2

`num2 = (15 % 7)`

Question 3

`num3 = (32 % 36)`

Question 4

`num4 = (5 - 3) + (10 - 5) * (8 % 3)`

Question 5

`num5 = 21 / 7 / 4 * (3 + 3)`

Question 6

`num6 = 9 / 3 + 21 - 5 * 2`

Question 7

`num7 = 7 % 5 + 6 * 2`

Question 8

`num8 = 35.2 / 2.3 + (332 % 33)`

Question 9

Add parentheses where needed

`num9 = 55 / 10 + 45 / 0.2`

`target9 = 5.0`

Question 10

Add parentheses where needed

`num10 = 65 // 20 + 10 - 4 % 4`

`target10 = 2`

Question 11

The following pseudocode describes how a bookstore computes the total price after buying a book.

```

Display "What is the price of the book?"
Get price of book
Display "How many copies of this book would you like?"
Get quantity of books
Compute the tax (6.5% of the total book price)
Compute the shipping charge ($1.95 per book)
Sum the total cost of the order
Display "The total cost of your order is: " and the total cost

```

Translate this pseudocode into a Python program.

Write your program for Question 11 in a file called `hw2_part2.py`.

Question 12

The following pseudocode describes how to extract the dollar and cents from a price when it is given as a floating-point value. For example, a price of 5.99 yields a value of 5 for the dollars and 99 for the cents.

```

Display "What is the price?"
Get price
Convert the price to an integer and store it in variable dollar.
Subtract the number of dollars from the price to get cents
Display "The number of dollars is: " and the number of dollars
Display "The number of cents is: " and the number of cents

```

Translate this pseudocode into a Python program.

Write your program for Question 12 in a file called `hw2_part3.py`.

Submitting

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

Don't forget to complete the header block for each file! If you need a reminder of what this should look like, see Lab 1. You should have a header block at the top of each file.

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

And remember, filenames are case sensitive.

You must be logged into your GL account, and you must be in the same directory as the `hw2_part1.py`, `hw2_part2.py`, and `hw2_part3.py` files. To double check this, you can use the `ls` command.

```
linux1[3]% ls
hw2_part1.py  hw2_part2.py  hw2_part3.py
linux1[4]% █
```

To submit your `hw2_part1.py`, `hw2_part2.py`, and `hw2_part3.py` files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW2`.

Type in the following command (on one line):

`submit cs201 HW2 hw2_part1.py hw2_part2.py hw2_part3.py`
and press enter.

```
linux1[4]% submit cs201 HW2 hw2_part1.py hw2_part2.py
hw2_part3.py
Submitting hw2_part1.py...OK
Submitting hw2_part2.py...OK
Submitting hw2_part3.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 was submitted by using the `submitls` command. Type in `submitls cs201 HW2` and hit enter. (You'll see your own username in place of "k38", of course.)

Double check to be sure that all three files have been submitted.

```
linux1[5]% submitls cs201 HW2
total 21
drwx-----  2 mneary1 rpc  2048 Aug 30 19:01 .
drwx----- 507 mneary1 rpc 18432 Aug 30 16:15 ..
-rw-r--r--   1 k38      rpc   167 Sep  6 19:01 hw2_part1.py
-rw-r--r--   1 k38      rpc    89 Sep  6 19:01 hw2_part2.py
-rw-r--r--   1 k38      rpc   102 Sep  6 19:01 hw2_part3.py
linux1[6]% █
```

And you're done!