

CMSC 201 Section 60

Homework 1 – Print, Input, Variables

Assignment: Homework 1
Due: Wednesday, September 16th at 11:59:59PM
Value: 50 points

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

This assignment falls under the standard cmsc201 academic integrity policy. This means you should not discuss/show/copy/distribute your solutions, your code or your main ideas for the solutions to any other student.

Instructions

For each of the questions below, you are given a program expressed in simple pseudocode. (This is similar to the in-class exercises we did during Lecture 2.) From this pseudocode, you must implement a working program in Python. For this exercise, you will only need to use concepts we have discussed in class such as variables, expressions, `input()`, casting to an integer or float, and `print()`.

The pseudocode may combine multiple lines of code into one step, or they may split something that would take a single line of code into multiple pieces. Think carefully about what the overall goal of the program is before you begin coding.

Although your programs should run without error, if you can not finish a certain part of this homework, submit whatever you have completed for partial credit. Do not assume that you will receive a zero simply because your code does not perfectly replicate the output.

Additional Instructions – Creating the hw1 Directory

During the semester, you'll want to keep your different Python programs organized, organizing them in appropriately named folders (also known as directories).

For Homework 1, you can store all six of the files you'll be creating in a single directory. First, navigate to the **Homeworks** directory inside the **cmsc201** directory (you can do this in a single “**cd**” command, as shown below). Next, create a directory to hold your Homework 1 files, and finally go into it.

```
linux3[1]% cd cmsc201/Homeworks
linux3[2]% mkdir hw1
linux3[3]% cd hw1
linux3[4]% █
```

From here, you can use **emacs** to start creating and writing your different Homework 1 Python programs.

You don't need to make a separate folder for each file. You should store all of the Homework 1 files in the same **hw1** folder.

Coding Standards

Coding standards for CMSC 201 can be [found here](#).

For now, you should pay special attention to the sections about:

- Naming Conventions
- Use of Whitespace
- Comments (specifically, File Header Comments)

We will not grade “harshly” on Coding Standards this early in the semester, but you should start forming good habits now. Make sure to pay attention to your TA’s feedback when you receive your Homework 1 grade back.

Input Validation

For this assignment, you do not need to worry about any “input validation.”

If the user enters a different type of data than what you asked for, your program may crash. This is acceptable.

If the user enters “bogus” data (for example: a negative value when asked for a positive number), this is acceptable. (Your program does not need to worry about correcting the value or fixing it in any way.)

For example, if your program asks the user to enter a whole number, it is acceptable if your program crashes if they enter something else like “dog” or “twenty” or “88.2” instead.

Here is what that error might look like:

```
Please enter a number: twenty
Traceback (most recent call last):
  File "test_file.py", line 10, in <module>
    num = int(input("Please enter a number: "))
ValueError: invalid literal for int() with base 10: 'twenty'
```

Questions

Question 1 (5 points)

Write your program for Question 1 in a file called `hw1_part1.py`.

You will write a small program which creates variables of a few different types and then outputs the total number of pets you have.

Translate this pseudocode into a Python program.

```
Create a variable named num_dogs
Create a variable named num_cats
Create a variable named num_fish
Create a variable named favorite_restaurant and set it to
    be the name of your favorite restaurant (you
    will be graded on whether it exists and is a
    string).
Create a variable golden_ratio and set it to be the
    golden ratio at least up to three decimal
    points.
```

Using those variables:

- 1.) Print out what your favorite restaurant is using the variable `favorite_restaurant`.
- 2.) Print out your approximation of the golden ratio using your variable `golden_ratio`.
- 3.) Print out the number of dogs, cats, and fish you've had.
- 4.) Print out the sum of pets you own.

(Yours does not have to match this word for word, but it should be similar. Do not worry about singular vs. plural)

```
linux4[116]% python3 hw1_part1.py
My favorite restaurant is Amber Spice
The golden ratio is about 1.618
I have 1 dogs, 0 cats, and 3 fish.
In total I have 4 animals.
```

Question 2 (5 points)

Write your program for Question 2 in a file called `hw1_part2.py`.

Translate this pseudocode into a Python program.

Ask the user for a Pokemon name.

Ask the user for a Pokemon type.

Ask the user for a Pokemon move.

Print out a sentence which combines the three pieces of information.

For this program, the names of the variables are not given to you. You should choose meaningful variable names. You are **not** to do any rounding, so whatever the division results in should be printed.

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

(Yours does not have to match this word for word, but it should be similar. The calculated costs should be the exact same, given the same input.)

```
linux4[118]% python3 hw1_part2.py
Pokemon name: Candytart
Pokemon type: Fairy
Pokemon move: Student Aid
New Gen9 Pokemon: Fairy type Candytart with the
special ability Student Aid!
```

Question 3 (10 points)

Write your program for Question 3 in a file called `hw1_part3.py`.

You are going to compute how much it costs to keep an animal for a year based on monthly food and other expenses.

Translate this pseudocode into a Python program.

```
Ask the user for the type of animal and store it in
variable pet_type
Ask the user how much they spend on food per month and
store in monthly_food
Ask the user how much they spend on other supplies per
month and store in monthly_supplies
Add these two variables together and then multiply by 12,
store result in annual_cost
Use the variables to print "The annual cost of owning an
animal_name is annual_cost."
```

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

(Yours does not have to match this word for word, but it should be similar.
The calculated costs should be the exact same, given the same input.)

```
linux4[122]% python3 hw1_part3.py
What type of pet do you own? Dog
How much do you spend per month on food? 75
How much do you spend per month on other supplies? 24
The annual cost of owning a Dog is $1188

linux4[123]% python3 hw1_part3.py
What type of pet do you own? Bird
How much do you spend per month on food? 20
How much do you spend per month on other supplies? 10
The annual cost of owning a Bird is $360

linux4[125]% python3 hw1_part3.py
What type of pet do you own? Seal
How much do you spend per month on food? 200
How much do you spend per month on other supplies? 100
The annual cost of owning a Seal is $3600
```


Question 4 (10 points)

Write your program for Question 4 in a file called `hw1_part4.py`.

Translate this pseudocode into a Python program.

You're working as a car salesman at your local dealership, where you make a 25-percent commission on gross profit. Gross profit is the profit a company makes after deducting the costs associated with making and selling its products. The average gross profit for this dealership is 5% of all car sales.

Write a program that calculates your commission amount given the car price and gross profit.

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

(Yours does not have to match this character for character but it should be nearly identical. White spaces are not important.)

```
linux4[127]% python3 hw1_part4.py
Please enter the car price: 20000
You made a 250.0 dollar commission on that car sale!

linux4[128]% python3 hw1_part4.py
Please enter the car price: 35000
You made a 437.5 dollar commission on that car sale!
```

Question 5 (10 points)

Write your program for Question 5 in a file called `hw1_part5.py`.

Translate this pseudocode into a Python program.

The energy contained in an object of rest mass m at velocity v , according to the Lorentz/Einstein equation is:

$$E = \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}}$$

Ask the user a rest mass.

Ask the user for a velocity.

Calculate and display the energy.

c is the speed of light, or 299,792,458 m/s.

Remember that

For this program, the names of the variables are not given to you. You should choose meaningful variable names.

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

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

```
linux4[138]% python3 hw1_part5.py
Enter the rest mass in kg: 50
Enter the velocity in m/s: 10
The Lorentz Energy in the object of rest mass 50.0 and
velocity 10.0 is 4.4937758936840904e+18

linux4[139]% python3 hw1_part5.py
Enter the rest mass in kg: 2000
Enter the velocity in m/s: 0.005
The Lorentz Energy in the object of rest mass 2000.0
and velocity 0.005 is 1.7975103574736352e+20
```

Question 6 (10 points)

Write your program for Question 6 in a file called `hw1_part6.py`.

Translate this pseudocode into a Python program.

The amount of force between two gravitational bodies is:

$$\frac{Gm_1m_2}{d^2}$$

where d is the distance between them, m_1 and m_2 are the masses and:

$$G = 6.674 \times 10^{-11} \frac{m^3}{kg \cdot s^2}.$$

Ask the user for a mass for object 1.

Ask the user for a mass for object 2.

Ask the user for a distance between objects.

Calculate and display the amount of force between the two gravitational bodies.

For this program, the names of the variables are not given to you. You should choose meaningful variable names. Assume d will not be entered as zero.

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

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

```
linux4[138]% python3 hw1_part6.py
What is the mass of object 1, in kg?
59720000000000000000000000000000
What is the mass of object 2, in kg?
198900000000000000000000000000000
What is the distance in meters between the objects?
152600000000
The gravitational force between the two objects is:
3.4043305119303843e+22

linux4[139]% python3 hw1_part6.py
What is the mass of object 1, in kg? 1
What is the mass of object 2, in kg? 1
What is the distance in meters between the objects? 1
The gravitational force between the two objects is:
6.674e-11

linux4[140]% python3 hw1_part6.py
What is the mass of object 1, in kg? 1000000
What is the mass of object 2, in kg? 1000000
What is the distance in meters between the objects?
100
The gravitational force between the two objects is:
0.0066739999999999999
```

Submitting

NOTE: How to submit is covered in detail in Lab 0 and Homework 0. If you have not read those assignments yet, you should do so before completing this part of the homework.

Once each or all of your files are complete, it is time to turn them in with the `submit` command. (You may also turn in individual files as you complete them. To do so, only `submit` those files that are complete.)

You must be logged into your account on GL, and you must be in the same directory as your Homework 1 Python files. To double-check you are in the directory with the correct files, you can type `ls`.

```
linux1[3]% ls
hw1_part1.py  hw1_part3.py  hw1_part5.py
hw1_part2.py  hw1_part4.py  hw1_part6.py
linux1[4]% █
```

To submit your Homework 1 Python files, we use the `submit` command, where the class is `cmsc201`, and the assignment is `HW1`. Type in (all on one line):

```
submit cmsc201 HW1 hw1_part1.py hw1_part2.py hw1_part3.py
hw1_part4.py hw1_part5.py hw1_part6.py
[then press enter]
```

```
linux1[4]% submit cmsc201 HW1 hw1_part1.py hw1_part2.py
hw1_part3.py hw1_part4.py hw1_part5.py hw1_part6.py
Submitting hw1_part1.py...OK
Submitting hw1_part2.py...OK
Submitting hw1_part3.py...OK
Submitting hw1_part4.py...OK
Submitting hw1_part5.py...OK
Submitting hw1_part6.py...OK
linux1[5]% █
```

(continued on next page)

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 see what files were successfully submitted by running:

```
submitls cmsc201 HW1
```

For more information on how to read the output from `submitls`, consult with Homework 0. Double-check that you submitted your homework correctly, since **empty files will result in a grade of zero**.

Advice for Submitting Early and Often:

You can also submit one or two files at a time (a good idea to do as you complete each part, *hint! hint!*) simply by modifying the command. For example, if you wanted to turn in just parts 2 and 3, type in `submit cmsc201 HW1 hw1_part2.py hw1_part3.py` and press enter.

```
linux1[4]% submit cmsc201 HW1 hw1_part2.py hw1_part3.py
Submitting hw1_part2.py...OK
Submitting hw1_part3.py...OK
linux1[5]% █
```

If you're re-submitting, the system will ask that you confirm you want to overwrite each file; make sure that you confirm by typing "y" and hitting enter if you want to do so.

```
linux1[5]% submit cmsc201 HW1 hw1_part3.py
It seems you have already submitted a file named
hw1_part3.py.
Do you wish to overwrite? (y/n):
y
linux1[6]% █
```