
CMSC 201 Fall 2017

Homework 1 – Pseudocode to Code

Assignment: Homework 1 – Pseudocode to Code

Due Date: Friday, September 15th, 2017 by 8:59:59 PM

Value: 40 points

Collaboration: For Homework 1, collaboration is allowed. Make sure to consult the syllabus about the details of what is and is not allowed when collaborating. You may not work with any students who are not taking CMSC 201 this semester.

If you work with someone, remember to note their name, email address, and what you collaborated on by filling out the Collaboration Log.

You can find the Collaboration Log at <http://tinyurl.com/collab201-Fa17>.

Remember that all collaborators need to fill out the log each time; even if the help was only “one way” help.

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.

```
# File:      FILENAME.py
# Author:    YOUR NAME
# Date:      THE DATE
# Section:   YOUR DISCUSSION SECTION NUMBER
# E-mail:    YOUR_EMAIL@umbc.edu
# Description:
#           DESCRIPTION OF WHAT THE PROGRAM DOES
```

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.

At the end, your Homework 1 files must run without any errors.

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 four of the files you'll be creating in a single directory. First, navigate to the `Homeworks` directory inside the `201` 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 201/Homeworks
linux3[3]% mkdir hw1
linux3[4]% cd hw1
linux3[5]% █
```

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

Prior to this assignment, you should read the Coding Standards, available on Blackboard under “Assignments” and linked on the course website at the top of the “Assignments” page.

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.

Additional Specifications

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), 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

Each question is worth 4 points. Following the directions is worth 8 points.

Question 1

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

This program, shown in pseudocode, prints out information about how many animals you (the programmer) might have as pets.

Translate this pseudocode into a Python program.

```
Set a number value for a variable called numDogs
Set a number value for a variable called numCats
Set a number value for a variable called numFish
```

```
Print out how many dogs, cats, and fish you have as pets
Print out the total number of pets you have
```

Here is some sample output, using sample values for *my* pets. (The number of dogs, cats, and fish you have will probably be different.)

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

```
bash-4.1$ python hw1_part1.py
I have 2 dogs as pets
I have 1 cats as pets
I have 4 fish as pets

I have a total of 7 pets
```

Remember to enable Python 3 before running and testing your code:

```
scl enable python33 bash
```

Question 2

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

This program, shown in pseudocode, asks for information about how much your bill was, how many people are in your party, and calculates and prints out the amount each person needs to pay.

Translate this pseudocode into a Python program.

```
Ask the user how much their total bill was
Ask the user how many people are in their dining party
Calculate the cost per person and print it
```

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. The calculated costs should be the exact same, given the same input.)

```
bash-4.1$ python hw1_part2.py
How much was your total bill? 72.60
How many people are in your dining party? 5
Each person needs to pay $ 14.52

bash-4.1$ python hw1_part2.py
How much was your total bill? 182.37
How many people are in your dining party? 7
Each person needs to pay $ 26.052857142857142

bash-4.1$ python hw1_part2.py
How much was your total bill? 15.00
How many people are in your dining party? 4
Each person needs to pay $ 3.75
```

Question 3

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

This program, shown in pseudocode, asks the user for the name of their dog, and then prints out that the dog is a good dog.

Translate this pseudocode into a Python program.

```
Ask the user for the name of their dog and store it in a
variable called dogName
Print out that dogName is a good dog
```

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.)

```
bash-4.1$ python hw1_part3.py
What is the name of your dog? Spot
Spot is a good dog!

bash-4.1$ python hw1_part3.py
What is the name of your dog? Thor
Thor is a good dog!

bash-4.1$ python hw1_part3.py
What is the name of your dog? His Royal Highness,
Christopher Rupert
His Royal Highness, Christopher Rupert is a good dog!
```

Question 4

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

This program, shown in pseudocode, asks the user for information about a car they're buying, and calculates and prints out the total cost of the car.

Translate this pseudocode into a Python program.

```
Ask the user how much the car costs
Ask the user how much the warranty costs
Ask the user how much the fees are
Ask the user how much the taxes are
Calculate and print the total cost of the car
```

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. The calculated costs should be the exact same, given the same input.)

```
bash-4.1$ python hw1_part4.py
How much does the car cost? 24999.99
How much does the warranty cost? 1499.99
How much are the fees? 699.99
How much are the taxes? 1480.12
The total cost of the car will be 28680.090000000004

bash-4.1$ python hw1_part4.py
How much does the car cost? 11000
How much does the warranty cost? 689.75
How much are the fees? 350
How much are the taxes? 399.64
The total cost of the car will be 12439.39
```

Question 5

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

This program, shown in pseudocode, asks the user for information about a restaurant, and prints out a summary of that information.

Translate this pseudocode into a Python program.

Ask the user for the restaurant's name, and store it in a variable called `nameInfo`

Ask the user for what type of food the restaurant serves, and store it in a variable called `foodInfo`

Ask the user for the restaurant's star rating, and store it in a variable called `ratingInfo`

Print out that `nameInfo` is a `ratingInfo` star restaurant, serving `foodInfo` food

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.)

```
bash-4.1$ python hw1_part5.py
What is the restaurant's name? Salty Dog
What food does the restaurant serve? pirate
What is the restaurant's star rating? 3
Salty Dog is a 3 star restaurant, serving pirate food

bash-4.1$ python hw1_part5.py
What is the restaurant's name? De Fina Hundarna
What food does the restaurant serve? Swedish
What is the restaurant's star rating? 5
De Fina Hundarna is a 5 star restaurant, serving
Swedish food
```


Question 6

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

This program, shown in pseudocode, asks the user for information about a bill they've received, and calculates and prints out how much they should pay in total.

Translate this pseudocode into a Python program.

```
Ask the user how much the bill was
Ask the user what the tax percentage is
Ask the user what percentage they want the tip to be
Calculate and print the total cost of the bill
```

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. The calculated cost should be the exact same, given the same input.)

```
bash-4.1$ python hw1_part6.py
How much was the bill? 100
How much is tax in your area? 6
What percentage do you want to tip? 15
The total cost of the meal will be 121.0

bash-4.1$ python hw1_part6.py
How much was the bill? 26.88
How much is tax in your area? 4.5
What percentage do you want to tip? 16.7
The total cost of the meal will be 32.578559999999996
```

Question 7

Write your program for Question 7 in a file called `hw1_part7.py`.

This program, shown in pseudocode, asks the user for their ratings about a restaurant they've eaten at, and prints out a summary of their experience.

Translate this pseudocode into a Python program.

Ask the user how the food was (from 1 to 5), and store it in a variable called `foodRating`

Ask the user how the service was (from 1 to 5), and store it in a variable called `servRating`

Ask the user how the atmosphere was (from 1 to 5), and store it in a variable called `atmoRating`

Calculate the average of the rating, and store it in a variable called `overallRating`

Print out the `overallRating` of the restaurant

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 ratings should be the exact same, given the same input.)

```
bash-4.1$ python hw1_part7.py
How would you rate the food? (1-5) 4
How would you rate the service? (1-5) 4
How would you rate the atmosphere? (1-5) 4
Your overall experience was 4.0 stars

bash-4.1$ python hw1_part7.py
How would you rate the food? (1-5) 4.9
How would you rate the service? (1-5) 3.67
How would you rate the atmosphere? (1-5) 4.18
Your overall experience was 4.25 stars
```

Question 8

Write your program for Question 8 in a file called `hw1_part8.py`.

This program, shown in pseudocode, asks the user for information about how many items of food they want to buy at a fixed price, and calculates and prints out the total cost of the meal.

Translate this pseudocode into a Python program.

```
Use variables to set number values for the cost of
appetizers, main dishes, and desserts
Ask the user how many appetizers, main dishes, and
desserts they want to order
Calculate and print the total cost of the meal
```

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

Here is some sample output, using sample cost values for *my* example restaurant. (The price of your appetizers, mains, and desserts will probably be different.) The user input is shown 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 and dish costs.)

```
bash-4.1$ python hw1_part8.py
Appetizers cost 9.5
How many appetizers would you like to order? 3
Main dishes cost 15.8
How many main dishes would you like to order? 8
Desserts cost 6.55
How many desserts would you like to order? 5

The total cost of your meal will be 187.65
```

Submitting

NOTE: How to submit is covered in detail in Lab 1 and Homework 0. If you have not read those assignments yet, you should do so before completing this part of the homework.

Once your `hw1_part1.py`, `hw1_part2.py`, `hw1_part3.py`, `hw1_part4.py`, `hw1_part5.py`, `hw1_part6.py`, `hw1_part7.py`, and `hw1_part8.py` 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_part7.py
hw1_part2.py  hw1_part4.py  hw1_part6.py  hw1_part8.py
linux1[4]% █
```

To submit your Homework 1 Python files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW1`. Type in (all on one line) `submit cs201 HW1 hw1_part1.py hw1_part2.py hw1_part3.py hw1_part4.py hw1_part5.py hw1_part6.py hw1_part7.py hw1_part8.py` and press enter.

(continued on next page)

```
linux1[4]% submit cs201 HW1 hw1_part1.py hw1_part2.py  
hw1_part3.py hw1_part4.py hw1_part5.py hw1_part6.py  
hw1_part7.py hw1_part8.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  
Submitting hw1_part7.py...OK  
Submitting hw1_part8.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 check that your homework was submitted by following the directions in Homework 0. Double-check that you submitted your homework correctly, since **an empty file will result in a grade of zero for this assignment.**