

---

## CMSC 201 Spring 2016

### Homework 5 – Algorithm Practice

**Assignment:** Homework 5 – Algorithm Practice

**Due Date:** **Tuesday**, March 8th, 2016 by 8:59:59 PM

**Value:** 40 points

Homework 5 is designed to help you practice using all of the structures we have learned so far including `for` loops, `while` loops, lists, iterating over lists, and branching selection structures. More importantly, you will be solving problems using algorithms you create and code yourself.

Remember to enable Python 3 before you run your programs:

```
scl enable python33 bash
```

### Instructions

In this homework, you will be doing a series of exercises designed to help you practice using `while` and `for` loops, control statements like `if/else`, `print()` statements, 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 should be commenting your code, and using constants in your code (not magic numbers or strings). Re-read the 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 5 is broken up into four parts. **Make sure to complete all 4 parts.**

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

And remember, filenames are case sensitive.

### hw5\_part1.py

(Worth 8 points)

Write a program to help calculate the tip at a restaurant, given the total bill and the level of service by the waiter. The user can enter that the service was “excellent”, “good”, or “bad”. If the user enters something else, you should re-prompt until they choose “excellent”, “good”, or “bad”.

The tip is based on the quality of the service with a minimum of \$2:

Excellent = 25% of the total

Good = 20% of the total

Bad = 10% of the total

You should print out the bill, the service level, the tip, and the grand total.

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

```
bash-4.1$ python hw5_part1.py
What is the total bill? 52.88
What was the level of service?
(Please choose excellent, good, or bad): fair
What was the level of service?
(Please choose excellent, good, or bad): good
The bill was $52.88
The service was good
The tip is $7.932
The grand total with tip is $60.812000000000005

bash-4.1$ python hw5_part1.py
What is the total bill? 11.99
What was the level of service?
(Please choose excellent, good, or bad): bad
The bill was $11.99
The service was bad
The tip is $2
The grand total with tip is $13.99
```

## hw5\_part2.py

(Worth 8 points)

Next you will write a program that tells you the cost of making copies at the UMBC copy center, which uses a sliding scale for pricing copies.

The first 100 copies are \$0.08 per copy.

The next 900 copies are \$0.06 per copy.

The next 9000 copies are \$0.05 per copy.

Any copies after the first 10,000 are \$0.04 per copy.

Make sure that you use constants, and not magic numbers, in your program!

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

```
bash-4.1$ python hw5_part2.py
How many copies do you want? 75
The total cost for 75 copies is $6.0

bash-4.1$ python hw5_part2.py
How many copies do you want? 240
The total cost for 240 copies is $16.4

bash-4.1$ python hw5_part2.py
How many copies do you want? 8000
The total cost for 8000 copies is $412.0

bash-4.1$ python hw5_part2.py
How many copies do you want? 1000
The total cost for 1000 copies is $62.0

bash-4.1$ python hw5_part2.py
How many copies do you want? 12345
The total cost for 12345 copies is $605.8
```

---

**hw5\_part3.py****(Worth 8 points)**

For this exercise, you are going to take in a phone number from the user, and print it back out in a standard format.

You can assume that the user will always enter the phone number in one of the three following ways (with no spaces):

1. (410)555-1212
2. 410-555-1212
3. 4105551212

Write a program that, given any of the three formats above, will output the number in the format “(410) 555-1212” (with a space between the closing parentheses and the following number). You may use casting, `input()`, for loops, `range()`, `len()`, and `if/elif/else` for this problem.

**You may not use any of Python's other built-in functions.**

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

```
bash-4.1$ python hw5_part3.py
Please enter the phone number: 4105551212
The formatted phone number is: (410) 555-1212

bash-4.1$ python hw5_part3.py
Please enter the phone number: (410)867-5309
The formatted phone number is: (410) 867-5309

bash-4.1$ python hw5_part3.py
Please enter the phone number: 800-555-2368
The formatted phone number is: (800) 555-2368
```

**hw5\_part4.py**

**(Worth 12 points)**

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

Your last task is to write a program that allows students to compete for extra credit points. In the points challenge, your program will alternate between two students, asking for the number of extra credit points that they would like to take. They can only take 1, 2, or 3 points.

The game begins by asking the first player how many points they would like to challenge for (13-21 points). If the first player chooses less than 13 or more than 21, it gives them an error and asks again.

After the first player chooses the number of points to play for, the second player “takes” 1, 2 or 3 points. If the player tries to take less than 1 or greater than 3 points, it gives them an error and asks again.

Then it is the first player’s turn, and they must “take” 1, 2 or 3 points.

Play continues until the winner takes the last point. At the end, the player may not “take” more points than are left. (In other words, if there are 2 points left, a player may not “take” 3 points. Your output must also be updated to show that the user may only take between 1 and 2 points.)

Your program should allow two users to play this game. At the end, it should print out the winner (either Player 1 or Player 2).

Sample output can be found on the next page.

*HINT:* You may need to use nested `while` loops for this program. Think carefully about how you want to design your program to keep track of who is playing, how many points are left, and what the players are allowed to do.

Here is some sample output for hw5\_part4.py, with the user input in blue.

```

bash-4.1$ python hw5_part4.py
Welcome to the Extra Credit Challenge!!!
Enter the number of points to play for (13-21): 22
Error: You can play for 13-21 points
Enter the number of points to play for (13-21): 0
Error: You can play for 13-21 points
Enter the number of points to play for (13-21): 16

Player 2 starts
There are 16 points left.
Player 2, how many points do you take? (1-3): 3

There are 13 points left.
Player 1, how many points do you take? (1-3): 0
Error: You can take 1-3 points
Player 1, how many points do you take? (1-3): 2

There are 11 points left.
Player 2, how many points do you take? (1-3): 4
Error: You can take 1-3 points
Player 2, how many points do you take? (1-3): 3

There are 8 points left.
Player 1, how many points do you take? (1-3): 2

There are 6 points left.
Player 2, how many points do you take? (1-3): 1

There are 5 points left.
Player 1, how many points do you take? (1-3): 3

There are 2 points left.
Player 2, how many points do you take? (1-2): 3
Error: You can take 1-2 points
Player 2, how many points do you take? (1-2): 2

Congratulations! Player 2 has won!!!

```

## Submitting

Once all four parts of your Homework 5 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 5 files. To double check this, you can type `ls`.

```
linux1[3]% ls
hw5_part1.py  hw5_part2.py  hw5_part3.py  hw5_part4.py
linux1[4]% █
```

To submit your files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW5`. Type in (all on one line)  
`submit cs201 HW5 hw5_part1.py hw5_part2.py hw5_part3.py hw5_part4.py`  
and press enter.

```
linux1[4]% submit cs201 HW5 hw5_part1.py hw5_part2.py
hw5_part3.py hw5_part4.py
Submitting hw5_part1.py...OK
Submitting hw5_part2.py...OK
Submitting hw5_part3.py...OK
Submitting hw5_part4.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 four homework files were submitted** by using the `submitls` command. Type in `submitls cs201 HW5` and hit enter.

And you're done!