

CSCI 331: Object Oriented Software Development

Lab #1 (100 points)

Due: Wednesday, January 21 by 23:59

Directions: The following lab will help you explore how to write Python programs using object-oriented software development concepts. Please label your programs `q<num>.py`, where `num` is the question number. For example, your solution to the first question, will be stored in the file `q1.py`.

Your answers should be based on the content and procedures covered in class, and not on other implementations or descriptions that can be found online.

Your programs should be driven by logic. You will not receive credit if you use statements to jump out of a loop: `break`, `continue`, etc.

Take your time and make sure you understand everything in this document before getting started. Make sure your programs match the output **EXACTLY** as given for each question. This is important, as one of the keys to being a good programmer is attention to details. Finally, don't leave things to the last minute.

1 Questions

1. Write a program (called `q1.py`) that finds the path with maximum sum for a set of values. For example, by starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is 23.

```
1      3
2     7 4
3    2 4 6
4   8 5 9 3
```

That is, $3 + 7 + 4 + 9 = 23$.

Find the maximum total from top to bottom for the values provided in `t100.txt`, a file containing a triangle with one-hundred rows.

It is not possible to use a brute force approach and try every route to solve this problem: there are 2^{99} possible combinations. If you could check one trillion (10^{12}) routes every second it would take over twenty billion years to check them all. One of the objectives of this lab is for you to find an efficient algorithm to solve this problem.

Example #1. At the prompt, the user provides `t4.txt` as a command line argument (line 1). The maximum total is reported on line 4.

```
1 python3 q1.py t4.txt
2 *****
3 *****
4 23
5 Elapsed time is 0.009508 seconds.
6 *****
```

Example #2. At the prompt, the user provides `t100.txt` as a command line argument (line 1). The maximum total is reported on line 4.

```
1 python3 q1.py t100.txt
2 *****
3 *****
4 7273
5 Elapsed time is 0.012927 seconds.
6 *****
```

Example #3. At the prompt, the user provides `foo.txt` as a command line argument (line 1). This argument corresponds to a file that does not exist. The program raises an exception and determines that an error occurred (line 2).

```
1 python3 q1.py foo.txt
2 ERROR: cannot open foo.txt
3 *****
4 *****
5 0
6 Elapsed time is 0.000596 seconds.
7 *****
```

2 Submitting Your Assignment

Once you have completed your program, submit it (`q1.py`) in VIU Learn in its corresponding Assignment page.