**Homework 9**

**Due: Sunday (see Syllabus)**

**Points: 40**

**Instructions**

In this exercise you will create a tiny interpreter for simple arithmetic operations. Your program will read it input from a file containing one command per line and it will write its output to a second file.

**Specifications**

1. Your program must ask the user for the names of both files. If there are problems with either file (inaccessible, unwritable, etc.) your program should complain and request file names again.
2. Your program should append “.txt” to a file name if it does not already have a .txt suffix.
3. The commands will look like this: operation value1 value2. See the Commands.txt file for details. If the input file does not exist or the output file is not writable the program must complain and ask the user for alternative file name(s).
4. The operations are **add**, **sub**, **multiply**, **divide**, and **power**. The commands are not case-sensitive. The values must be integers.
5. An empty line should be silently ignored.
6. Your program must read the commands from the input file one at a time, interpret them, and write results into the output file. Your program must stop when it reaches the end of the input file.
7. Your program must use exception handling to deal with exceptions such as divide-by-zero and improperly formed inputs (commands and values).
8. For your final submission your program must be run at least twice. Use **Commands.txt** for one run. Create at least one other input file for your other run(s).
9. Your submission must include your Python program, the input files, the output files, and your screen captures. Don’t do a screen capture of the input or output files – just provide them when you submit.

Run your program several times using different inputs – sufficient to demonstrate that your program meets all the assignment requirements. Capture a screen shot of each run and paste them into an MS Word document. Place a caption above each image.

**Submit the Python .py file (lastname\_hw9.py) containing your program and the MS Word document** **to your instructor using the appropriate Assignment Submissions link.**

Attach: Commands.txt

**Sample Output**

Text

Description automatically generated

**Contents of Commands.txt**

add 11 22

DIVIDE 99 11

Multiply 22 10001

…

**Contents of Output.xt**

Interpreter Runtime Output

1 : add(11,22) = 33

2 : divide(99,11) = 9.0

3 : multiply(22,10001) = 220,022

…