

COM S 1270 - 'Lucky Calculator!'

Please see the 'Course Schedule' on the Syllabus for this assignment's due date.

Assignment Objective

The goal of this assignment is to give you practice working with mathematical operators, functions, and the `random` module.

The activity for this assignment is for you to create your own 'calculator' program in a file called `luckyCalculator.py`.

Instructions

This 'calculator' should allow the user to choose either a basic math operation (+, -, *, /, //, %, **), or to choose to print out a 'lucky number.'

The calculator should also compensate for any 'divide by zero errors' in some way. This could mean printing an error message and immediately quitting the program, changing the operand that is a zero to be a one instead, etc. There are many possible equally valid solutions to this problem which affects the results produced by the /, //, and % operators. The only thing the program cannot do is crash as it would under normal circumstances when dividing by zero.

Additionally, the `luckyCalculator.py` script should output a 'header,' including the name of the program ("Lucky Calculator!"), your name, and the class/ section you are in as per the Example Output below.

Please see the example output below for details about how the calculator program should work.

HINT: To generate the 'lucky number,' try exploring Python's `random` module:

<https://docs.python.org/3/library/random.html>

HINT: You can get access to the `random` module by importing it at the top of your script:

```
import random
```

Please note, there are no loops/ iteration involved in this assignment at all. However, it will require the use of `if/ elif/ else` statements. Students are advised to explore the official Python documentation to gain a bit of familiarity with this concept:

<https://docs.python.org/3/tutorial/controlflow.html>

https://docs.python.org/3/reference/compound_stmts.html#if

<https://docs.python.org/3/reference/expressions.html#booleans>

Other Requirements:

Additionally, you do *not* need to compensate for the crashes that occur when the user enters a value of the incorrect type. (E.g., The program asks for an integer, but the user enters a letter instead.)

It is required for you to write your name, the date you started working on your script, and a short explanation of what your code does at the top of your file. For example:

```
# Matthew Holman          6-6-2024
# Assignment #2
#
# This basic calculator can do 7 different operations,
# and even generate a lucky number!
```

Do *not* attempt to 'double dip' by placing your 'header' printout information at the top of your file - the header and the initial comment block are different things.

Your work should be your own, completely original effort. Meaning - you should not just copy my code explanation above, but rather you should try to come up with one of your own.

The actual Python script itself can be programmed in any way (excluding cheating ala ChatGPT) so long as the output resembles that below. This includes the use of the initial 'header' when starting the program (i.e. the program title, who it is by, the student's class/ section, etc.). Please note that the 'header' will be strictly enforced with this assignment. Be sure to include the title, your name, and your class/ section.

Example Output

Lucky Calculator!

By: Matthew Holman
[COM S 127 A]

What would you like to do?

[c]alculator, [l]ucky number, [q]uit: c

Please Choose a Calculation [+], [-], [*], [/], [//], [%], [**]: +
Please Enter An Integer: 2
Please Enter An Integer: 3
The result of your calculation was: 5

Lucky Calculator!

By: Matthew Holman
[COM S 127 A]

What would you like to do?

[c]alculator, [l]ucky number, [q]uit: c

Please Choose a Calculation [+], [-], [*], [/], [//], [%], [**]: //
Please Enter An Integer: 5
Please Enter An Integer: 0
ERROR in // Function: b = 0
The result of your calculation was: 5

Lucky Calculator!

By: Matthew Holman
[COM S 127 A]

What would you like to do?

[c]alculator, [l]ucky number, [q]uit: l

Please Enter An Integer: 1
Please Enter An Integer: 10
Your lucky number is: 9

Lucky Calculator!

By: Matthew Holman
[COM S 127 A]

What would you like to do?

[c]alculator, [l]ucky number, [q]uit: q

Goodbye!

Lucky Calculator!

By: Matthew Holman
[COM S 127 A]

What would you like to do?

[c]alculator, [l]ucky number, [q]uit: asdf

ERROR: I did not understand your input... Please try again...

Lucky Calculator!

By: Matthew Holman
[COM S 127 A]

What would you like to do?

[c]alculator, [l]ucky number, [q]uit: c

Please Choose a Calculation [+], [-], [*], [/], [//], [%], [**]: asdf
ERROR: You must enter either "+", "-", "*", "/", "//", "%", or "**"

Special Notes

NOTE: This assignment should not be terribly difficult. However:

- Completing this assignment may require you to start your work 'before the last minute.' Please plan accordingly.
- Your script **CANNOT** crash under any circumstances except for those circumstances noted below. Any portions of your code where the script crashes will receive a zero (0) for that aspect of the assignment.
 - Understand, when the word 'crash' is used in the instruction above, what this means is 'crashing under expected use cases given the level of knowledge attained in the class.'
 - You do not currently have the ability to enforce user input types (although you will later in the semester). As such, if you ask for an integer as input (e.g., 1), and the user provides a letter (e.g., 'a'), and it crashes, then this is *not* a problem *yet*.

NOTE: You are turning in your CODE - NOT the OUTPUT of your code.

- Your code will be run by the TAs, and the output of those runs, along with the code itself, is what will be evaluated.

NOTE: Assignments turned in in any other format other the specified file types will not be accepted.

- Screenshots of code **will not** be accepted.
- .sln files are **not** code files - they contain **no** Python code and **will not** be accepted.
- .zip, .rar, .tar.gz, and other compressed files **will not** be accepted unless otherwise specified.
- If your submission is not in a .py file, when so specified, or if the submission is not accompanied by **all** the files needed to run the submission, the submission will not be graded.
 - **THIS WILL LEAD TO YOU RECEIVING A ZERO (0) ON THE ASSIGNMENT.**
 - **You will *NOT* be allowed to re-submit your work after the final deadline in this case.**

NOTE: You have the ability (and responsibility) to 'double check' that your work/ submission is correct when you turn it in.

- If you accidentally turn in the wrong submission, you will **not** be allowed to resubmit it after the final deadline.
- Please note, you can submit your work as many times as you like *before* the final deadline.