

Lab 13

Learning Objectives:

- III. **What broke and how do I fix it?** *How do I test code for errors?* This lab will give you the opportunity to practice using `UnitTest` to find unexpected results and errors in code, so that they can be fixed before anything bad happens.
- IV. **How do I communicate science and Python with others?** *How can I make code that other people can't break?* This lab will give you the opportunity to give your robustified code to someone else to see if they can break it.
- F. **Write and Test, Write and Test....** *Write code in small bits that you can easily test.*

In **part 1** you'll write and test a naive Python calculator.

In **part 2** you'll trade yours with a classmate and try to break each other's code.

Part 1: Building a Robust Calculator

We've provided you with a naive Python calculator. It can do simple things, but it gets confused if you give it something too complicated. **Your task is to robustify it.**

The calculator is implemented in `calc.py`. This program is designed to be run from the command line with, for example,

```
./calc.py 1+1
2
```

But if you do something a bit more complicated,

```
./calc.py 10+5
None
```

it will return something unexpected!

Your task is to sequentially come up with test cases that `calc.py` *should* pass and then make it pass them.

Write the tests using `unittest` in `testing.py`. Some ideas are below.

- Implement and test a help message for incorrect syntax.
- Implement and test subtraction, multiplication, and division.
- Implement and test multi-digit numbers.
- Implement and test floating-point numbers.
- Implement and test multiple additions/subtractions or multiplications/divisions in a row.
- Implement and test quantities that have units: `./calc.py 10 m + 5 m` should give `15 m`.

Part 2: Break Your Partner's Code

Find a group of two or three and have each person give his or her code to someone else. Now try to break the code you were given: think of things you can enter that your partner didn't take into account!