

# Exercise 1

## Due June 28, 12:00pm

It is finally your time to code! This exercise will start off at a slow pace. Be sure to follow exact instructions as your work is going to be put through an auto-marker (that has absolutely no sympathy). Feel free to use GitHub to save your progress.

- In the folder that this exercise is in, you will be seeing a rubric.
- You do get marks for merging your branch into dev-branch and then merging into master.
- You get a mark for following the PEP8 style guide. You can check your styled work by going onto <http://pep8online.com/>
- I will manually check that you do follow the design recipe
  - However, you cannot use the examples I provided on this pdf. If you do, deductions will be made.
- There will be deductions if you submit your file with a print statement, an input statement, or an infinite loop, or an exact description of what I put on here.
- Feel free to make functions to help you. I won't mark them.

In a file called **exercise\_1.py** you will be writing and submitting the following 3 functions:

### Function 1: square\_me

Create a function called **square\_me** that takes in a float and returns the square of that number as a result. Examples:

```
>>> square_me(4.0)
16.0
>>> square_me(113.5)
12882.25
```

### Function 2: kanye\_helper

Kanye West travelled back in time, met Paul McCartney, and wrote out lines of lyrics. Kanye finally figured out the name of his song **All Day**. Unfortunately, Kanye does not have enough time to answer every single question he put into the song, with the lyrics " All Day.", and wants a function to do it. You can safely assume that each string inputted will at least have one character. Create a function that answers the string's question, if it exists. You can also safely assume that a question is a string that ends with a question mark.

```
>>> kanye_helper("How long do these dudes ball?")
"How long do these dudes ball? All Day."
>>> kanye_helper("Shoppin in the winter and it's just May")
"Shoppin in the winter and it's just May"
```

### Function 3: `course_mark`

Sometimes determining your course mark can be a hassle, so this function is a skeletal version of how you can calculate it. This table is taken from my Operations class:

Content	Out of	Weight
Assignment 1	10	15
Assignment 2	20	15
Midterm	42	20
Final	4	50

The mark is calculated by:

$$mark = \sum_{i=1}^n \frac{\text{Mark Received}_i}{\text{Out Of}_i} \cdot \text{Weight}_i$$

Create a function called `course_mark` that takes in 4 integers and returns a float of my calculated mark for my operations class. You can safely assume that any mark I insert will not be above the threshold and will not be below 0.

**Hint: You can use a fixed global variable(s) and a helper function**

```
>>> course_mark(7, 14, 33, 1)
49.214285714285715
>>> course_mark(1, 11, 41, 4)
79.27380952380952
```