Instructions: Please answer the following questions making sure to write your answers legibly on the second page. The code below is the contents of python file that we will work with.

- Since we are writing python code please be careful about spacing / indent. Leave enough room that it is obvious when you want things indented.
- Make sure to write legibly.
- If you are unfamiliar with the `time` module, `time.time()` returns an integer representation of time, subtracting two of these objects, as we do in the code below will return the number of elapsed seconds.
- Similarly, `time.sleep()` when provided an integer will "pause" the program for that many seconds.

```
import time

def timing_decorator(func):
    def wrapper(*args, **kwargs):
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(f'QUESTION 1: {kwargs}')
        print(f"{func.__name__}")
        print(f"took {end - start} seconds")
        return result
    return wrapper


def add_two(a, b):
    return a + b

def add_three(a, b, c):
    return a + b + c


@timing_decorator
def calc_function(var1, var2, var3='Processor', var4=1):
    print(f"Calc Type: {var3}, number: {var4}")
    time.sleep(var1 + var2)


if __name__ == "__main__":
    calc_function(1, 2, var3='computer', var4=3)
```

---

1. The decorator above reports the time that it takes for the function to complete. The line

   `print(f'QUESTION 1:  kwargs')`

   will print something to the terminal.

   (a) In this line of code what <u>type</u> of python object does `kwargs` behave like?

   (b) When the code is executed what will be printed to the terminal by this line (only return what will be printed by this specific line)?

2. In the second f-string in the decorator is the phrase `func.__name__`. If the code above is run, what will be printed here?

3. The functions `add_two` and `add_three` are defined above. Please write a function (`F`) which accepts either function and arguments. It should send those arguments to the function specified and return the result of that function multiplied by two.

   In other words `F(add_two, 1, 2)` should return $2 \cdot (1 + 2) = 6$ and `F(add_three, 1, 2, 3)` should return $2 \cdot (1 + 2 + 3) = 12$.