Objects as parameters and return types
Objects can be passed to functions as arguments and also returned from functions.

Ex1: time.py

```python
# File: time.py

class Time:
    '''Represents the time of day.

    attributes:
    hour: int, minute: int, second: int
    '''

def print_time(t: Time):
    print( ('%.2d:%.2d:%.2d') % ( t.hour, t.minute, t.second) )

def time_in_seconds(t: Time) -> int:
    '''Returns the time in seconds.''
    return t.hour * 3600 + t.minute * 60 + t.second

def time_from_seconds(seconds: int) -> Time:
    '''Returns a Time object given the time in seconds.''
    time = Time()
    time.hour, seconds = divmod(seconds, 3600)
    time.minute, time.second = divmod(seconds, 60)
    return time

def is_after(t1: Time, t2: Time) -> bool:
    ''' Returns true if t1 is after t2 .''
    return (t1.hour, t1.minute, t1.second) > (t2.hour, t2.minute, t2.second)
```

Last updated: February 23, 2020 at 11:31 AM
def add_seconds(t: Time, sec: int) -> Time:
    '''Return a new Time with the added seconds.'''
    sec += time_in_seconds(t)
    return time_from_seconds(sec)

def add_time(t1: Time, t2: Time) -> Time:
    '''Returns a Time object that is the sum of the two parameter objects.'''
    total_seconds = time_in_seconds(t1) + time_in_seconds(t2)
    return time_from_seconds(total_seconds)

t1 = Time()
t1.hour = 10
t1.minute = 30
t1.second = 5

print('t1 = ', end = '')
print_time(t1)

t2 = Time()
t2.hour = 9
t2.minute = 40
t2.second = 30

print('t2 = ', end = '')
print_time(t2)

t3 = add_seconds(t2, 40)
print('t3 = ', end = '')
print_time(t3)

print(f'Is t1 > t2? {is_after(t1, t2)}')