

BIS 420 PROGRAMMING FOR DATA SCIENCE
PRAJAKTA POHARE
CHAPTER 16 EXERCISE 16.3
ILLINOIS STATE UNIVERSITY

Write a correct version of increment that doesn't contain any loops. Anything that can be done with modifiers can also be done with pure functions. In fact, some programming languages only allow pure functions. There is some evidence that programs that use pure functions are faster to develop and less error-prone than programs that use modifiers. But modifiers are convenient at times, and functional programs tend to be less efficient.

In general, I recommend that you write pure functions whenever it is reasonable and resort to modifiers only if there is a compelling advantage. This approach might be called a functional programming style.

```
class Time:
```

```
    def __init__(self, hour=0, minute=0, second=0):
```

```
        self.hour = hour
```

```
        self.minute = minute
```

```
        self.second = second
```

```
def increment(time, seconds):
```

```
    total_seconds = time.hour * 3600 + time.minute * 60 + time.second + seconds
```

```
    new_hour = total_seconds // 3600
```

```
    remaining = total_seconds % 3600
```

```
    new_minute = remaining // 60
```

```
    new_second = remaining % 60
```

```
    return Time(new_hour, new_minute, new_second)
```

```
t1 = Time(1, 59, 30)
```

```
t2 = increment(t1, 90)
```

```
print('%0.2d:%0.2d:%0.2d' % (t2.hour, t2.minute, t2.second))
```

```
class Time:
    def __init__(self, hour=0, minute=0, second=0):
        self.hour = hour
        self.minute = minute
        self.second = second

def increment(time, seconds):
    total_seconds = time.hour * 3600 + time.minute * 60 + time.second + seconds

    new_hour = total_seconds // 3600
    remaining = total_seconds % 3600
    new_minute = remaining // 60
    new_second = remaining % 60

    return Time(new_hour, new_minute, new_second)

t1 = Time(1, 59, 30)
t2 = increment(t1, 90)

print('%0.2d:%0.2d:%0.2d' % (t2.hour, t2.minute, t2.second))
```