

Class 16 - Notes

Upcoming Schedule

Before Wednesday's class, either (1) read [Chapter 7: Cost](#) of the course book, or (2) complete [Lesson 5: How Programs Run](#) of the Udacity course. (After Wednesday, you should do the other one.) The example programs in Chapter 7 of the coursebook are in Scheme (not Python), but you should be able to mostly understand them (and it's good practice to be reading Scheme code, since you'll be writing a Scheme interpreter soon!). If you have questions about any of the code examples, please let me know.

Project 4 will be posted soon, and due March 16.

Code

class16.py

Notes and Questions

```
def list_map(fn, lst):
    def appender(lst, e):
        return lst + [fn(e)]
    return list_accumulator(appender, lst, [])

def list_map(fn, lst):
    return list_accumulator(lambda lst, e: lst + [fn(e)],
                           lst, [])
```

What does it *cost* to compute?

Why does it cost more per instruction to compute in Tokyo than Virginia?

Why would someone pay a premium to do their computing in Tokyo?

How expensive is addition of whole numbers?

How expensive is multiplication of whole numbers?

Links

Black belts (and wanna-be black belts) will also want to look at how Python actual implements addition and multiplication in [longobject.c](#).

Fourth-degree black belts should read Fürer's paper: Martin Fürer, *Faster Integer Multiplication*. STOC 2007.