Announcements

Marking ongoing! Grades for both A1 and the midterm should be released early next week.
A Bit More on Recursion

CSC148, INTRODUCTION TO COMPUTER SCIENCE
DAVID LIU
def nested_list_contains(obj, item) -> bool:
    if isinstance(obj, int):
        return obj == int
    else:
        for sublist in obj:
            if nested_list_contains(sublist, item):
                return False
A common error: missing return

def nested_list_contains(obj, item) -> bool:
    if isinstance(obj, int):
        return obj == int
    else:
        for sublist in obj:
            if nested_list_contains(sublist, item):
                return False

return False
A common error: missing `return`

def nested_list_contains(obj, item) -> bool:
    if isinstance(obj, int):
        return obj == int
    else:
        for sublist in obj:
            if nested_list_contains(sublist, item):
                True
        return False
A common error: missing `return`

A return statement exits from *one function call*.

When writing a recursive function that should return something, both the base case and recursive step must have a return!

More generally, if a function returns something, then every execution path through the function must have a return.
“Source code is a liability, not an asset.”
Simplifying list creation

A typical for loop.

```
data = []
for x in input:
    data.append(f(x))
```

A list comprehension.

```
data = [f(x) for x in input]
```
Simplifying list creation

A typical for loop.

```python
data = []
for x in input:
    data.append(f(x))
```

A list comprehension.

```python
data = [f(x) for x in input]
```