

Function As Argument Activity

Imagine that you have a list of lists of integers. Sort it according to...

- The number of items per list
- The median of each list

For example, if L contains...

```
L = [[5, 4, 3], [700], [4, 7, 1, 3]]
```

Then...

```
[[700], [5, 4, 3], [4, 7, 1, 3]] # sorted by length. Hint: use len()
[[5, 4, 3], [4, 7, 1, 3], [700]] # sorted by median. Hint: write a helper to
                                # calculate the median
```

Decorator Activity

Write a function decorator for logging. When added to a function, it prints out the arguments to the calling function as well as the return. Consider the following example:

```
@logging
def myFunc(x, y):
    return x+y

v = 0
for i in range(2):
    for j in range(2):
        v += myFunc(i, j)
print(v)
```

Prints...

```
Calling myFunc(0, 0)
returned: 0
Calling myFunc(0, 1)
returned: 1
Calling myFunc(1, 0)
returned: 1
Calling myFunc(1, 1)
returned: 2
4
```

```
def noneInsteadOfEmpty(f):
    def g(*args):
        L = f(*args)
        if len(L) == 0:
            return None
        else:
            return L
    return g
```