

## 6.00 Handout, Lecture 12

(Not intended to make sense outside of lecture)

```
def readVal(valType, requestMsg, errorMsg):
    while True:
        val = raw_input(requestMsg)
        try:
            val = valType(val)
            return val
        except ValueError:
            print(errorMsg)
```

```
def getGrades(fname):
    try:
        gradesFile = open(fname, 'r')
    except IOError:
        raise ValueError('Could not open ' + fname)
    grades = []
    for line in gradesFile: grades.append(float(line))
    return grades
```

```
try:
    grades =
    getGrades('q1grades.txt')
    grades.sort()
    median = grades[len(grades)/2]
    print 'Median grade is', median
except ValueError, err:
    print 'Whoops.', err
```

```
class Person:
    def __init__(self, family_name, first_name):
        self.family_name=family_name
        self.first_name=first_name
        self.age = None
    def familyName(self):
        return self.family_name
    def firstName(self):
        return self.first_name
    def setAge(self, age):
        if type(age) == int or type(age) == float:
            self.age = age
        else:
            raise TypeError
    def __gt__(self, other):
        return self.family_name + self.first_name >\
            other.family_name + other.first_name
    def __str__(self):
        return self.first_name + ' ' + self.family_name
```

```
me = Person('Guttag', 'John')
print me.firstName()
print me
her = Person('Hockfield', 'Susan')
him = Person('Obama', 'Barack')
print me > him
print me > her
print him > her
people = [him, her, me]
people.sort()
for p in people:
    print p
```