Generators

- Any procedure or method with a `yield` statement is called a generator

```
def genTest():
  yield 1
  yield 2
```

- `genTest() ⇒ <generator object genTest at 0x201b878>`
- Generators have a `next()` method which starts/resumes execution of the procedure. Inside of generator
  - `yield` suspends execution and returns a value
  - Returning from a generator raises `StopIteration` exception

Fix to Grades class

Before

```
def allStudents(self):
    if not self.isSorted:
        self.students.sort()
        self.isSorted = True
    # return copy of student list
    return self.students[:]
```

After

```
def allStudents(self):
    if not self.isSorted:
        self.students.sort()
        self.isSorted = True
    for s in self.students:
        yield s
```