Magic Methods for Custom Classes

```
class Vector():
    def __init__(self, elems):
        self.elems = elems

    def size(self):
        return len(self.elems)

v = Vector([1,2])
len(v)  # => fails
```

```
class Vector():
    def __init__(self, elems):
        self.elems = elms

    def __len__(self):
        return len(self.elems)

v = Vector([1,2])
len(v)  # => succeeds
```

Avoid using Catch-Alls

```
while True:
    try:
        n = int(input("> "))
    except:
        print("Invalid input.")
    else:
        return n ** 2
```

```
while True:
    try:
        n = int(input("> "))
    except ValueError:
        print("Invalid input.")
    else:
        return n ** 2
```