

## Separation of Concerns

• Game ending conditions

A design principle: Isolate different parts of a program that address different concerns A modular component can be developed and tested independently

Hog	Hog Game Simulator	Game Commentary	Player Strategies	
	•Game rules	• Event descriptions	<ul> <li>Decision rules</li> <li>Strategy parameters         (e.g., margins &amp;             number of dice)</li> </ul>	
	<ul><li>Ordering of events</li><li>State tracking to determine the winner</li></ul>	•User input		
Ants	Ants Game Simulator	Actions	Tunnel Structure	
	<ul><li>Order of actions</li><li>Food tracking</li></ul>	• Characteristics of different	• Entrances & exits • Locations of insect	

Example: Restaurant Search

## Restaurant Search Data

```
Given the following data, look up a restaurant by name and show related restaurants.

{"business_id": "gclB3ED6uk6viWlolSb_uA", "name": "Cafe 3", "stars": 2.0, "price": 1, ...}

{"business_id": "WXKx2I2SEzBpeUGtDMCS8A", "name": "La Cascada Taqueria", "stars": 3.0, "price": 2}
...

{"business_id": "gclB3ED6uk6viWlolSb_uA", "user_id": "xVocUszkZtAqCxgWak3xVQ", "stars": 1, "text": "Cafe 3 (or Cafe Tre, as I like to say) used to be the bomb diggity when I first lived in the dorms but sadly, quality has dramatically decreased over the years....", "date": "2012-01-19", ...}

{"business_id": "WXKx2I2SEzBpeUGtDMCS8A", "user_id": "84dCHkhWG8IDtk30VvaY5A", "stars": 2, "text": "-Excuse me for being a snob but if I wanted a room temperature burrito I would take one home, stick it in the fridge for a day, throw it in the microwave for 45 seconds, then eat it. NOT go to a resturant and pay like seven dollars for one...", "date": "2009-04-30", ...}
...
```

(Demo)

**Example: Similar Restaurants** 

## Discussion Question: Most Similar Restaurants

Implement **similar**, a **Restaurant** method that takes a positive integer **k** and a function **similarity** that takes two restaurants as arguments and returns a number. Higher **similarity** values indicate more similar restaurants. The **similar** method returns a list containing the **k** most similar restaurants according to the **similarity** function, but not containing **self**.

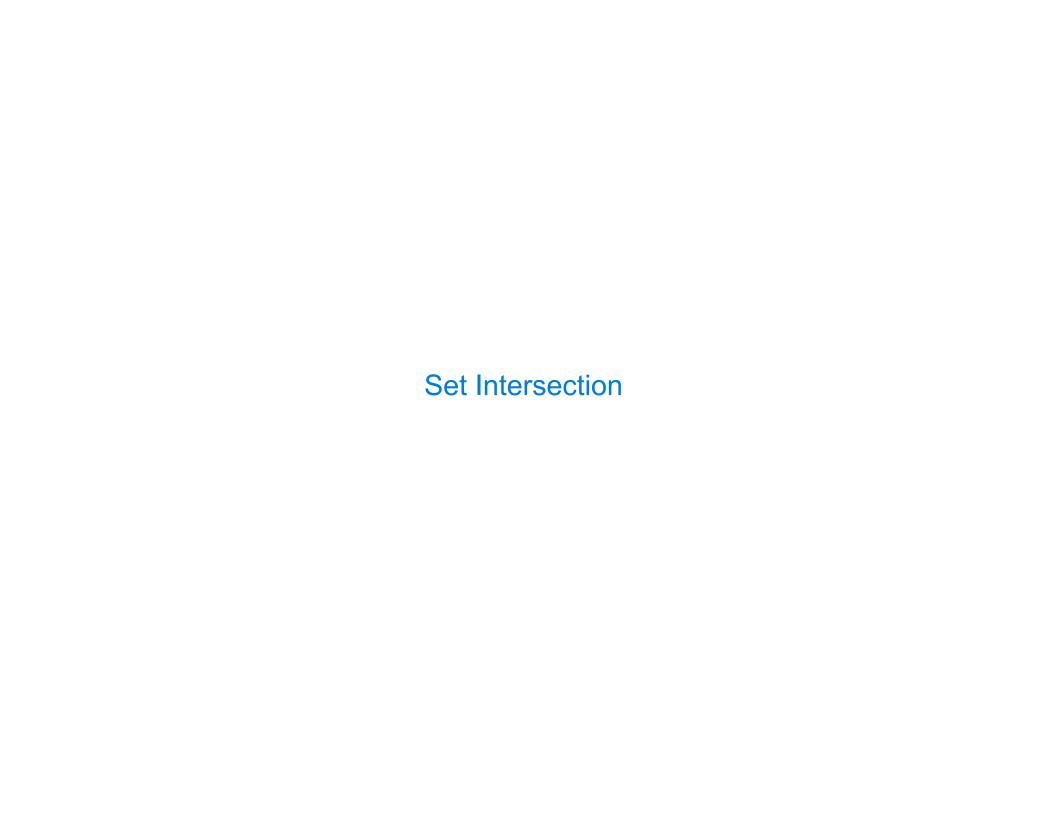
```
def similar(self, k, similarity):
    "Return the K most similar restaurants to SELF, using SIMILARITY for comparison."
    others = list(Restaurant.all)
    others.____remove (____self___)
    return sorted(others, key=___lambda r: -similarity(self, r) ___)__[:k]
```

sorted(iterable, /, \*, key=None, reverse=False)
Return a new list containing all items from the iterable in ascending order.
A custom key function can be supplied to customize the sort order, and the

reverse flag can be set to request the result in descending order.

**Example: Reading Files** 

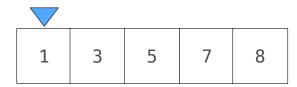
(Demo)



## **Linear-Time Intersection of Sorted Lists**

Given two sorted lists with no repeats, return the number of elements that appear in both.

3	4	6	7	9	10



(Demo)

return count