

LECTURE 19 — CLASSES, PART 2

35.1 Overview

- Review of classes
- Revisiting our Yelp data: a `Restaurant` class.
- Techniques that we will see:
 - Calling class methods from within the class
 - Class objects storing other objects, such as lists
 - Lists of class objects

35.2 Review of Classes

We will use our `Point2d` class solution from Lecture 18 to review the following:

- Attributes:
 - These store the data associated with each class instance.
 - They are usually defined inside the class to create a common set of attributes across all class instances.
- Initialization: function `__init__` called when the object is created.
 - Should assign initial values to all attributes
- Methods
 - Each includes the object, often referred to as `self`, as the first argument.
 - Some change the object, some create new objects
- Special methods start and end with two underscores. Python interprets their use in a variety of distinct ways:
 - `__str__` is the string conversion function
 - `__add__`, `__sub__`, etc. become operators
- Each of these special methods builds on the “more primitive” methods

35.3 Larger Example — Restaurant Class

Recall Lab 5 on the Yelp data:

- Read and parse input lines that look like:

```
The Greek House|42.73|-73.69|27 3rd St+Troy, NY 12180|\
http://www.yelp.com/biz/the-greek-house-troy|Greek|1|5|4|5|4|4|5|5|5|5|4
```

- Find restaurants and print out information based on a user selection
- Original implementation based on a list was awkward:
 - We had to remember the role of each index of the list — 0 was the name, 1 was the latitude, etc.
- New implementation here is based on a class

35.4 Start to a Solution, the Main Code

Let's look at `lec19_restaurants_exercise.py`, downloadable as part of the `Lecture_19` zip file:

- This is the code that *uses* the `Restaurant` class.
 - We start by considering how the class will be used rather than how we write it.
 - Main function to initialize a restaurant is called `convert_input_to_restaurant`
 - Parses a restaurant line
 - Creates and returns a `Restaurant` object
 - Function `build_restaurant_list`
 - Opens the input file
 - Reads each line
 - Calls `convert_input_to_restaurant`, and appends the resulting restaurant to the back of a list
 - Main code:
 - Builds the restaurant list
 - Prints the first three restaurants in the list
 - Includes commented-out code that
 - * Gets the name of a city
 - * Finds the restaurant with the highest average rating
- We will complete this code soon.

35.5 Functionality Needed in the Restaurant Class

- Some functionality is determined by reading the code we have already written
 - Includes both methods and attributes
- Add other functionality by considering the methods that must be in the `Restaurant` class, including the parameters that must be passed to each method.

- Add attributes last...

35.6 Turning to the Actual Restaurant Class

Look at `Restaurant.py` which was distributed with the `Lecture_19` files.

- The `__init__` function specifies the attributes.
 - Other attributes could be added, such as the average rating, but instead these are computed as needed by methods.
 - Importantly, each class object stores a list of ratings, illustrating the fact that classes can store data structures such as lists, sets, and dictionaries.
- The `Restaurant` class has more complicated attributes than our previous objects
 - `Point2d` object,
 - A list for the address entries
 - A list of scores
- There is nothing special about working with these attributes other than they “feel” more complicated.
 - Just apply what you know in using them
 - Our lecture exercises will help

35.7 In-Class Example

Together we will add the following two methods `Restaurant` to get our demonstration example to work:

1. The `is_in_city` method
2. The `average_review` method

35.8 Discussion

- What is not in the `Restaurant` class?
 - No input or line parsing. Usually, we don’t want the class tied to the particular form of the input.
 - As an alternative, we could add a method for each of several different forms of input.
- Often it is hard to make the decision about what should be inside and what should be outside the class.
 - One example the method we wrote to test if restaurant is in a particular city. As an alternative we could have written a different method that returns that name of the city and make the comparison outside the class.
- We could add an `Address` class:
 - Reuse for objects other than restaurants
 - Not needed in this (relatively) short example.
 - More flexible than our use of a list of strings from an address line.

35.9 Summary

- Review of the main components of a Python class:
 - Attributes
 - Methods
 - Special methods with names starting and ending with `__`
 - * Initializer method is most important
- Important uses of Python classes that we have seen today:
 - Classes containing other objects as attributes
 - Lists of class objects.
- Design of Python classes
 - Start by outlining how they are to be used
 - Leads to design of methods
 - Specification of attributes and implementation of methods comes last