

# Computer Science 1 — CSci 1100

## Lab 4 — Yelp Data

### Spring Semester 2012-2013

#### Lab Overview

This lab explores more complex use of lists, logic, and file input, all through the example of analyzing real data provided by Yelp for restaurants near RPI.

In order to get started, please create a folder in your Dropbox for Lab 4 and download the zip file `yelp.zip` from the Piazza Resources page into this new folder. Unzip it to find two text files, `yelp-short.txt` and `yelp.txt`. The first is just a shortened version of the second. You will use both data files to test your code.

#### Preliminary Work

Take a look at `yelp-short.txt` and in particular at the first line, which is

```
Meka's Lounge|42.74|-73.69|407 River Street+Troy, NY 12180|http://www.yelp.com/biz/\
mekas-lounge-troy|Bars|5|2|4|4|3|4|5
```

The line includes the name of the restaurant, the latitude and longitude of the restaurant, the street address of the restaurant, the URL for the Yelp information about the restaurant, the type of restaurant (`Bars`, in this case), and some number of ratings in the range 1-5.

We need to get this information into a Python program in a form we can use. Create a Python file in your Lab 4 folder called `read_file.py` which has the following code

```
in_file = open('yelp-short.txt','r')
# in_file = open('yelp.txt','r')

for line in in_file:
    print line[0], len(line)
```

This code

- opens `yelp-short.txt` and assigns it to a variable called `in_file`,
- reads each line of the file loop into a string called `line`, and
- prints the first character and length of the line.

We gave you similar code with HW 3.

Go ahead and run this code. You should get 10 lines of output. If you want to read `yelp.txt` just add the `#` to the start of the first line and remove it from the second. After you do this and rerun the program, you will get 155 lines of output. In Lecture 14 we will discuss input files in more detail.

We are now ready for the actual checkpoints.

## Checkpoint 1

Copy `read_file.py` to `check1.py`. In `check1.py`, write a function called `parse_line` that takes as its sole parameter the input line. It should return a list containing the relevant information from the input line. For example, for the first line of input, `parse_line` should return the list

```
["Meka's Lounge", "42.74", "-73.69", "Bars", [ '5', '2', '4', '4', '3', '4', '5' ] ]
```

Note that entry 4 of the list is itself a list. The other entries are, in order, the name, the latitude, the longitude, and the type of restaurant. For now, all of them should be strings.

Your main code should be modified to look like

```
in_file = open('yelp-short.txt','r')
# in_file = open('yelp.txt','r')

for line in in_file:
    p_line = parse_line(line)
    print p_line
```

In detail, `parse_line` should

- apply `strip('\n')` to remove the end of line character,
- split the resulting string into a list variable (I'll call it `L0`) using character `'|'`,
- create the final list to return from entries 0, 1, 2, 5, and `6:len(L0)` of `L0`

Run this code only on the shorter input file, since the longer file generates too much output.

**To complete Checkpoint 1**, show the lab TA or a mentor the code and the output.

## Checkpoint 2

Before getting started, copy `check1.py` to `check2.py`.

In this checkpoint you need to improve your parsing so that the latitude and longitude are float values, the list of reviews is a list of ints, and the final entry in the list returned by `parse_line` is the average review score. For example the first line should look like

```
["Meka's Lounge", 42.74, -73.69, "Bars", [5, 2, 4, 4, 3, 4, 5], 3.857142857142857 ]
```

Please go ahead and do this now, using the main function from the end of Checkpoint 1 to test. You will need to think about the list functions and type conversions to apply.

**To complete Checkpoint 2**, show the lab TA or a mentor the code and the output when running on just `yelp-short.txt`.

## Checkpoint 3

Before getting started, copy `check.py` to `check3.py` and continue to work on `check3.py`.

Modify your code to ask the user for the type of restaurant (via `raw_input`) and a minimum rating. Then, modify the `for` loop to check the list that results from parsing to see if the

restaurant matches the type of restaurant the user is interested in and has a minimum rating. If so, output the name of the restaurant and the rating of the restaurant, accurate to 2 decimal points. For example, here is the output from our solution when applied to the full yelp.txt

```
What type of restaurant would you like => Italian
What is the minimum rating => 3.6
Name: Spiaks Restaurant; Rating 3.86
Name: Lo Porto's; Rating 4.00
Name: Verdile's Restaurant; Rating 4.00
Found 3 restaurants.
```

The final output is the number restaurants found during this search.

**To complete Checkpoint 3**, demonstrate your code and your output finding Pizza restaurants and finding Italian restaurants.