

I210: Information Infrastructure I – Mastery Check 1

When you have completed this Mastery Check, you should:

- 1. Submit your solution to Canvas under "Mastery Check 1"*
- 2. Upload a copy of your work to OneDrive as a backup in case something goes wrong with Canvas.*

Make sure you click through to the final "Submit" button on Canvas. You are responsible for making sure your work is submitted **by the end of the lab**, so it is strongly advised that you verify the submission before leaving. **We CANNOT accept late submissions.**

As a reminder, you may use anything on the Canvas section for **I210 only**, as well as the course textbook. You may use any notes (physical or online) taken for **I210 only**. You may work on an STC machine or on your personal laptop.

You may NOT use the Internet except:

- the Canvas sections for I210
- any group coding or note spaces you've set up for I210.

Using code you found online from outside of this class or code that you or your group did not write that is not from the book or slides is likely to constitute academic misconduct.

Overview:

You have found that it's hard to figure out which of your friends like going to specific restaurants with you. You start working on a program that will help you store data about restaurants, their prices, and which friends like which restaurants.

Using this data, you will answer some questions:

- When you pick a restaurant, which of your friends want to go there with you?
- How much will it cost to go to that restaurant?
- For specific pairing of your friends, do they have restaurant interests in common?

You will show the answers to all of these questions in your output.

SAMPLE FINISHED OUTPUT:

```
Restaurants liked by Amir: {'Avers', 'Mother Bears', 'Pizza X', 'Wendys', 'Taco Bell'}
Restaurants liked by Brett: {'Avers', 'Noodles', 'KFC', 'McDonalds', 'Taco Bell'}
Restaurants liked by Carmen: {'Pizza X', 'Wendys', 'Azzip', 'Kilroys', 'Qdoba'}
Restaurants liked by Davon: {'Pizza X', 'Wendys', 'Noodles', 'Little Tibet', 'Korea Restaurant'}
Restaurants liked by Eva: {'Pizza X', 'Texas Roadhouse', 'Noodles', 'Burger King', 'Little Tibet'}

The information I have about prices: {'Avers': '$$', 'Azzip': '$', 'Burger King': '$', 'KFC': '$', 'Kilroys': '$$', 'Korea Restaurant': '$$', 'Little Tibet': '$$', 'McDonalds': '$', 'Mother Bears': '$$', 'Noodles': '$', 'Pizza X': '$$', 'Qdoba': '$$', 'Taco Bell': '$', 'Texas Roadhouse': '$$$', 'Wendys': '$'}

-----
Where would you like to eat? Taco Bell
Amir also likes Taco Bell
Brett also likes Taco Bell
It will cost $ to eat there.

-----
The restaurants that Amir and Brett both like are: {'Avers', 'Taco Bell'}
They should expect to pay: $$ or $

There is only one restaurant Carmen and Eva both like. It is: {'Pizza X'}
They should expect to pay: $$

There are no restaurants that Brett and Carmen both like. set()
```

Don't forget to add comments as you code!

Section 1 – Enter the Data (30 points):

Download [Mastery Check 1 Starter.py](#)

Most of the data you need is already here for you, but you need to finish this section by adding the data for the last friend, Eva.

- Step 1a: Add one more variable for Eva and list the restaurants they like (see below). Be sure to use the restaurants listed in the image below.
- Step 1b: Output the restaurants liked by each friend as shown below.
- Step 1c: Manually type in any restaurant names and price ranges (“\$” – “\$\$\$”) associated with restaurants Eva likes that are not already included in the prices data.
- Step 1d: Once you have entered this data, output it to the user as shown below.

Note: All of the places where you need to add code for this step say “Step 1” in the comments. You will want to use the correct data types for each piece. Pay attention to how they are displayed in the example output below. If your output has different symbols, you’re probably using a different data type. **Note:** The width of your screen may affect how the output looks.

```
Restaurants liked by Amir: {'Mother Bears', 'Avers', 'Wendys', 'Pizza X', 'Taco Bell'}
Restaurants liked by Brett: {'McDonalds', 'KFC', 'Noodles', 'Avers', 'Taco Bell'}
Restaurants liked by Carmen: {'Wendys', 'Azzip', 'Kilroys', 'Qdoba', 'Pizza X'}
Restaurants liked by Davon: {'Little Tibet', 'Noodles', 'Wendys', 'Korea Restaurant', 'Pizza X'}
Restaurants liked by Eva: {'Texas Roadhouse', 'Little Tibet', 'Noodles', 'Burger King', 'Pizza X'}

The information I have about prices: {'Avers': '$$', 'Azzip': '$', 'Burger King': '$', 'KFC': '$',
'Kilroys': '$$', 'Korea Restaurant': '$$', 'Little Tibet': '$$', 'McDonalds': '$', 'Mother Bears':
 '$$', 'Noodles': '$', 'Pizza X': '$$', 'Qdoba': '$$', 'Taco Bell': '$', 'Texas Roadhouse': '$$$',
 'Wendys': '$'}
```

Section 2 – User Input (30 points):

- Step 2a: Ask the user where they want to eat. (see below)
 - Please note: Because this is a user input, we won’t know what restaurant will be entered. Due to that, we will need to utilize the user input variable in steps 2b and 2c. We cannot assume that it will always be “Taco Bell” and hardcode in the responses. We need to use the variable in comparison to the list items and use the variable as a key for the dictionary.
- Step 2b: Check to see if that restaurant name is in the list of restaurants for each friend. Print out a message for each friend who likes to eat there.
- Step 2c: Also print out what it costs to eat there.
- Example output:

```
-----
Where would you like to eat? Taco Bell
Amir also likes Taco Bell
Brett also likes Taco Bell
It will cost $ to eat there.
```

If the restaurant that the user inputs is not contained in any of your friends' lists, it would be nice to have a response indicating so.

Remember that you could combine lists using sets (Lesson 3) in order to compare values or utilize compound comparisons (Lesson 6) using and / or / in / not in

If none of your five friends like the restaurant you enter, display this:

```
Where would you like to eat? The Elm
Sadly, no one likes to eat at The Elm
```

Section 3 – Friend Overlap (30 points):

Now that you've compared your chosen restaurant to your friends' preferences, it's time to compare them to each other. We need to see where one friend's preferences intersects with the other. (Lesson 3)

- Step 3a: Should Amir and Brett want to go out to eat, where can they go? What do you they have in common? Where do their restaurants intersect? Once you know that, please use the restaurant names to find their price ranges so they will know much they will pay. Goal: Find the restaurants they both like and their price ranges.
- Step 3b: Should Carmen and Eva want to go out to eat, where can they go and how much will they pay? Find the restaurant (just one!) they both like and using its name, find its price range.
- Step 3c: Brett and Carmen want to go out to eat, but they have no common restaurants! Prove this using Python.

Reminder: `set()` is how we show the empty set in Python!

```
-----
The restaurants that Amir and Brett both like are: {'Avers', 'Taco Bell'}
They should expect to pay: $$ or $
```

```
There is only one restaurant Carmen and Eva both like. It is: {'Pizza X'}
They should expect to pay: $$
```

```
There are no restaurants that Brett and Carmen both like. set()
```

Section 4 – Comments (10 points)

Be sure that you have written in comments explaining your code. You could add comments to the end of each line, above or below each line or section. Where you do it is up to you.

Bonus Section (+10 points):

Create a set of all of the restaurants using the data between the five friends. Once you have that, have Python help you determine which restaurant occurs earliest (or lowest) in the alphabet using a Python function.

Note: When we say compute here, we mean you can't just hard-code the text – you must write python code to find the answer. Do not type out all of the restaurants.

Example output:

```
-----  
All of the restaurants: {'Avers', 'Pizza X', 'Azzip', 'Texas Roadhouse', 'KFC', 'Little Tibet',  
'Burger King', 'McDonalds', 'Korea Restaurant', 'Mother Bears', 'Wendys', 'Noodles', 'Kilroys',  
'Qdoba', 'Taco Bell'}
```

```
The one that occurs first is: Avers
```