Project 4: Classes “Restaurant”

This project will give you a chance to structure code in a large project.

PART I: (worth 20 points)

- Make a class called **Restaurant** (restaurant.py). The __init__method for **Restaurant** should store two attributes (restaurant_name, cuisine_type) and methods **describe_restaurant(), open_restaurant()** that prints a message indicating that the restaurant is open.
- Make an instance called restaurant from your class and display a message about the restaurant.

PART II: Number Served (worth 20 points)

- Add an attribute called **number_served** with a default value of 0. Create an instance called restaurant from this class. Print the number of customers the restaurant has served, change this value and print it again.
- Add a method called **set_number_served()** that lets you set the number of customers that have been served. Call this method with a new number and print the value again.
- Add a method called **increment_number_served()** that lets you increment the number of customers who’ve been served. Call this method with any number you like that could represent how many customers were served in, say, a day of business.

PART III: Ice Cream Stand: An ice cream stand is a specific kind of restaurant (worth 30 points)

Write a class called **IceCreamStand** (ice_cream_stand.py) that inherits from the Restaurant class you wrote in Part I, II. Add an attribute called flavors that stores a list of ice cream flavors. Write a method that displays these flavors. Create an instance of **IceCreamStand** that displays the Ice Cream Stand’s flavors.

PART IV: Imported Modules: (worth 30 points)

Write a Python program (my_restaurant.py) that imports **Restaurant** and **IceCreamStand** classes. You program should access the multiple classes.

Project Heading:

Use the following as a header for all of your projects:

```
#--------------------------------------------------------------------------------------------------------------
# Program name – filename.py
# Written by – your name
# Date – todays date
# Description of the program.
#--------------------------------------------------------------------------------------------------------------
```

Style:

- Keep of your code structure simple

- Class Names should be written in **CamelCaps.**
- Use the same formatting conventions use in functions.
- Each module should have a **docstring** describing what the classes in a module can be used for.
- Find an approach that lets you write coded that works, and go from there.

Due Date:

Week 15 - 16

Turn in:

1. Algorithm or flow chart
2. Include all Python files with your name and indicate is project 4. *e.g. restaurant_P4_YourName.py*