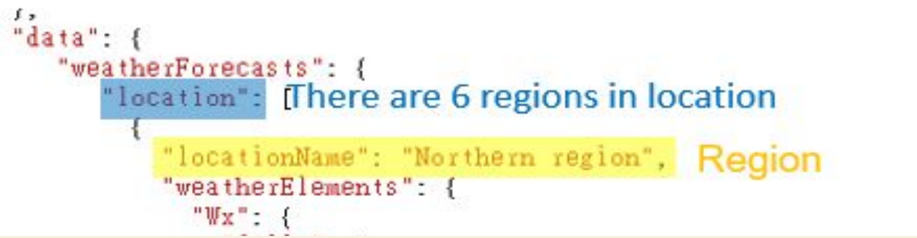
12.17 LAB

LAB HOUR: 18:30-21:30

EXERCISE 1

“`One week weather.json`” has weather information. Please use the object oriented concept to write a program that computes the average of minimum temperature in each region for one week, also get the region name and weather description of Sunday.

In the main function, you need to create a “RegionWeather” instance for each region by the constructor, then print regions’ name and the average of minimum temperature in that region(round off to the 2nd decimal place), also print the weather description of Sunday.



```
,
"data": {
   "weatherForecasts": {
      "location": [
         {
            "locationName": "Northern region",
            "weatherElements": {
               "Wx": {
```

There are 6 regions in location

Region

EXERCISE 1

Constraints :

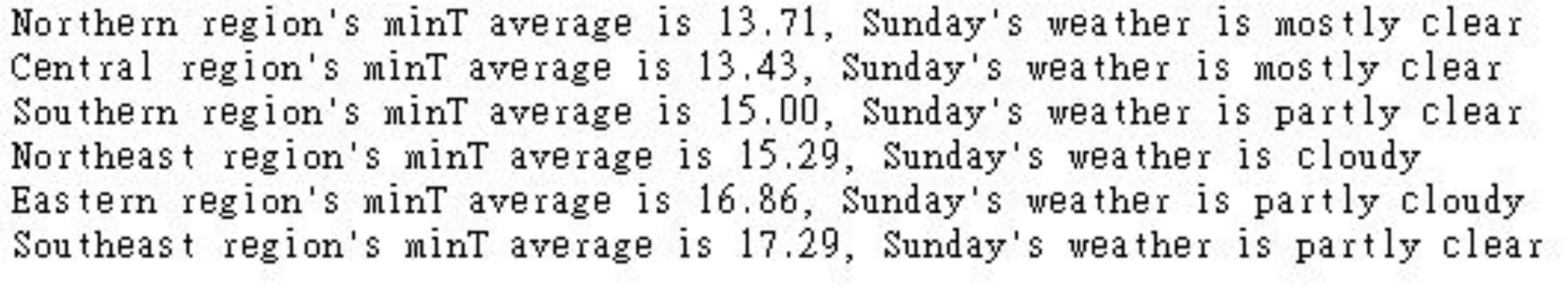
1. You need to create a class called “RegionWeather”. In the class, it must have three properties (highlights below):

“region_name” : locationName in .json
“sunday_weather_description” : Wx in .json of Sunday(2019-12-08)
“minT_average” : average of minimum temperature(MinT in .json) for one week
You must use @property (include getter & setter) to get and set these properties.

2. You must implement methods in class “RegionWeather” which compute values for each setter to set, NOT IN THE MAIN.
3. Output strings must be the same as the next slide.

EXERCISE 1

Output strings should be like:



```
Northern region's minT average is 13.71, Sunday's weather is mostly clear
Central region's minT average is 13.43, Sunday's weather is mostly clear
Southern region's minT average is 15.00, Sunday's weather is partly clear
Northeast region's minT average is 15.29, Sunday's weather is cloudy
Eastern region's minT average is 16.86, Sunday's weather is partly cloudy
Southeast region's minT average is 17.29, Sunday's weather is partly clear
```

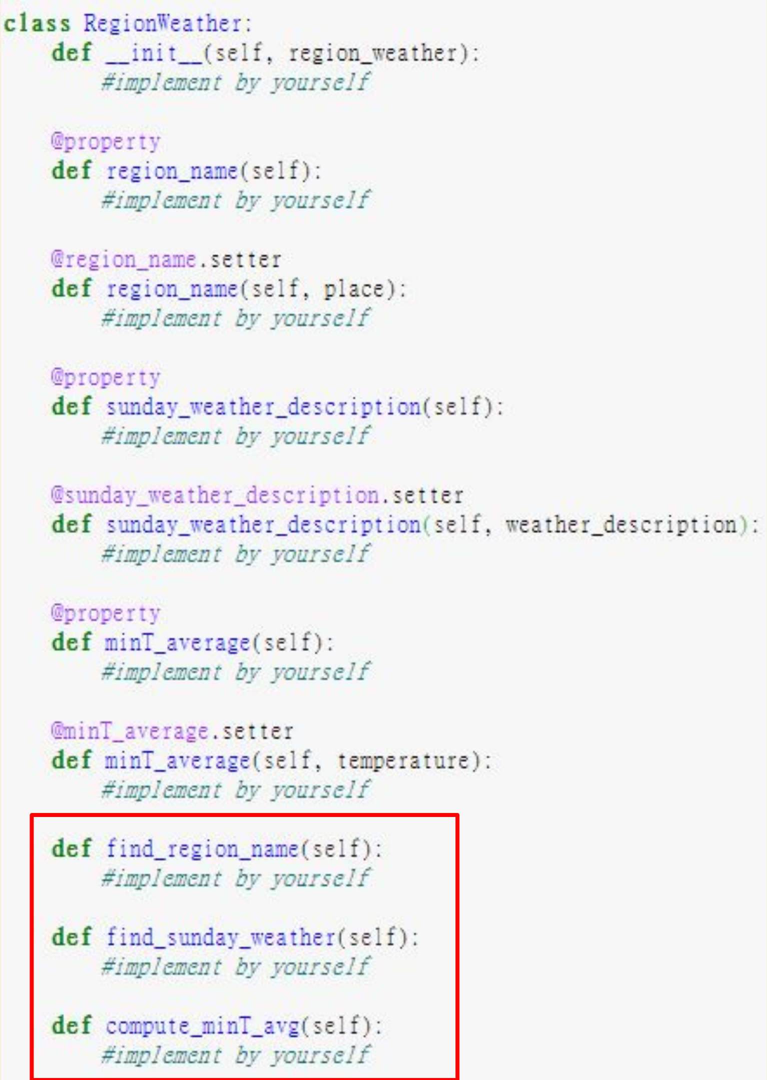
EXERCISE 1

The following two slides are the hint snapshots implemented by teaching assistant, they might be helpful to you for coming up with solutions to this exercise.

EXERCISE 1

class

Example code



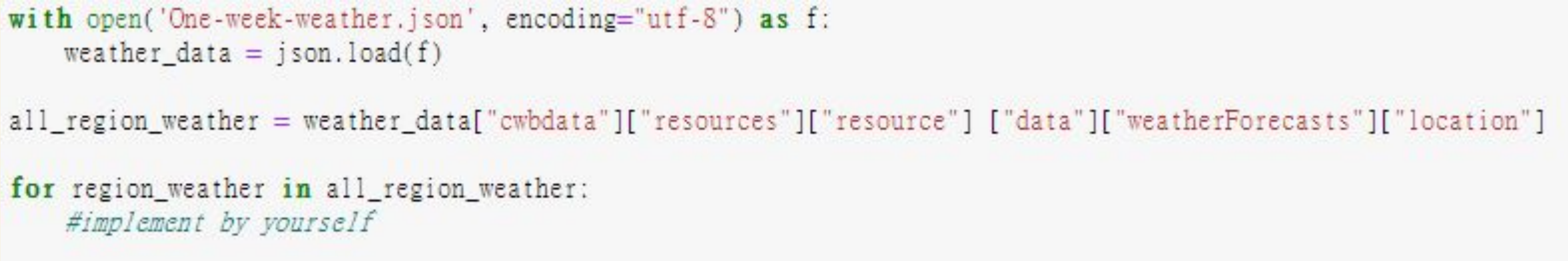
```
class RegionWeather:
    def __init__(self, region_weather):
        #implement by yourself

    @property
    def region_name(self):
        #implement by yourself

    @region_name.setter
    def region_name(self, place):
        #implement by yourself

    @property
    def sunday_weather_description(self):
        #implement by yourself

    @sunday_weather_description.setter
    def sunday_weather_description(self, weather_description):
        #implement by yourself

    @property
    def minT_average(self):
        #implement by yourself

    @minT_average.setter
    def minT_average(self, temperature):
        #implement by yourself

    def find_region_name(self):
        #implement by yourself

    def find_sunday_weather(self):
        #implement by yourself

    def compute_minT_avg(self):
        #implement by yourself
```

These methods are used for computing property values

EXERCISE 1

Example code

main



```python
with open('One-week-weather.json', encoding="utf-8") as f:
    weather_data = json.load(f)

all_region_weather = weather_data["cwbdata"]["resources"]["resource"] ["data"]["weatherForecasts"]["location"]

for region_weather in all_region_weather:
    #implement by yourself
```

EXERCISE 2

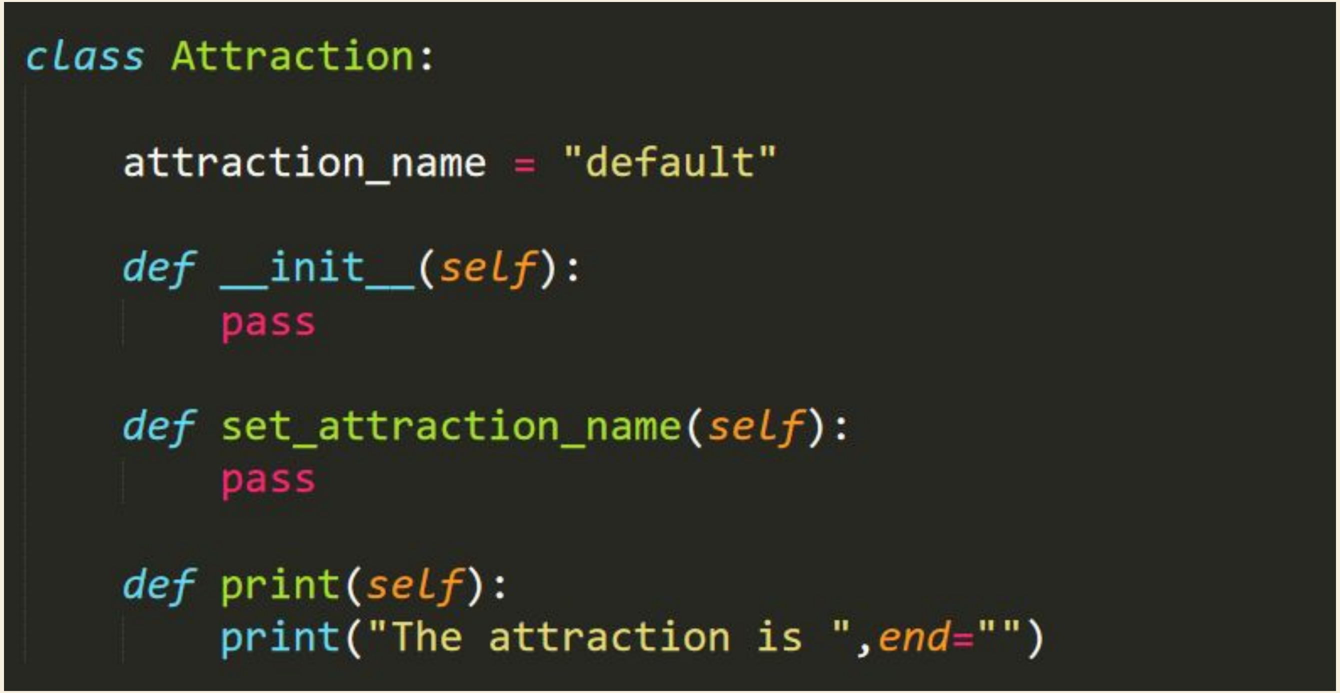
In EXERCISE 2, You have to use the concept of class inheritance.

When people mention Kaohsiung, they always think of Love River, the famous tourist attraction there. The definition of class "**Attraction**" is described below, try to create a subclass "**Place**" inherit from "**Attraction**" and construct three objects of it, set their attraction as "**Love River**","**Love Ferris Wheel**" and "**Disneyland**" respectively. Finally, print different strings based on the three attractions by the method defined in "**Place**"(you have to figure out the rules by your own).
Make sure that your code satisfied the constraints in the following four slides.

Hint : understand how to use "super()" might be helpful.

EXERCISE 2

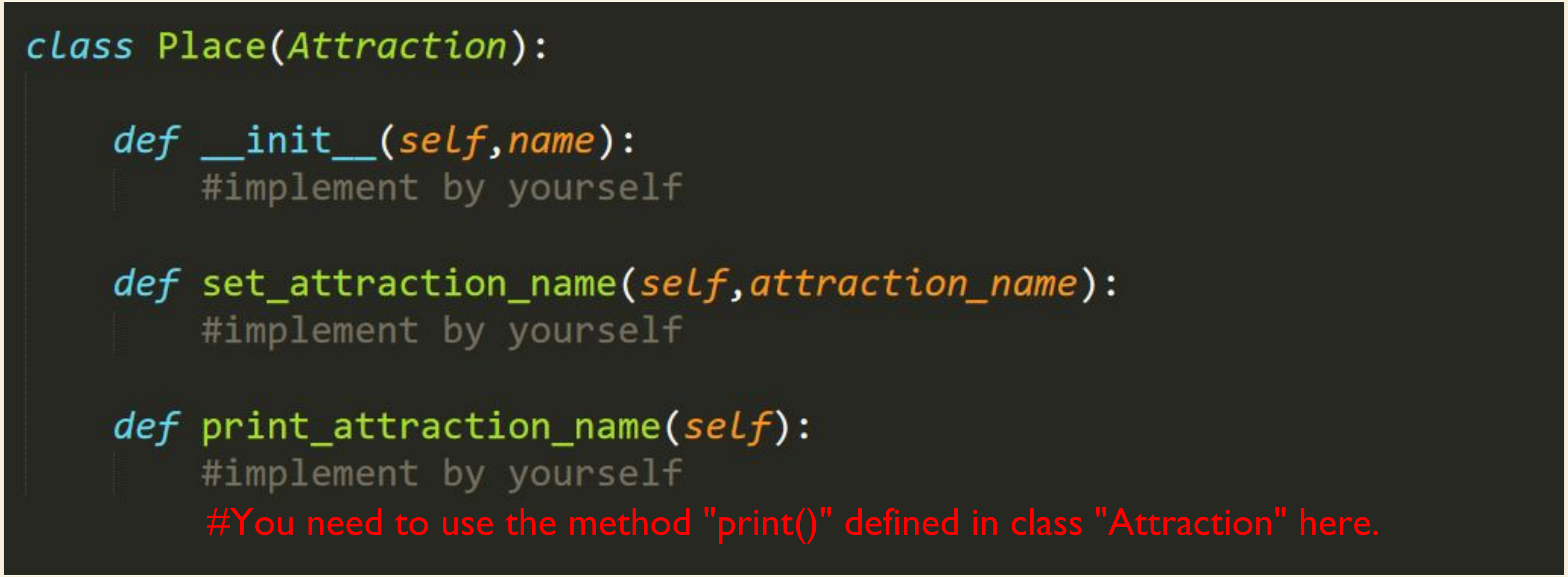
class **Attraction** must be defined as



```python
class Attraction:

    attraction_name = "default"

    def __init__(self):
        pass

    def set_attraction_name(self):
        pass

    def print(self):
        print("The attraction is ",end="")
```

EXERCISE 2

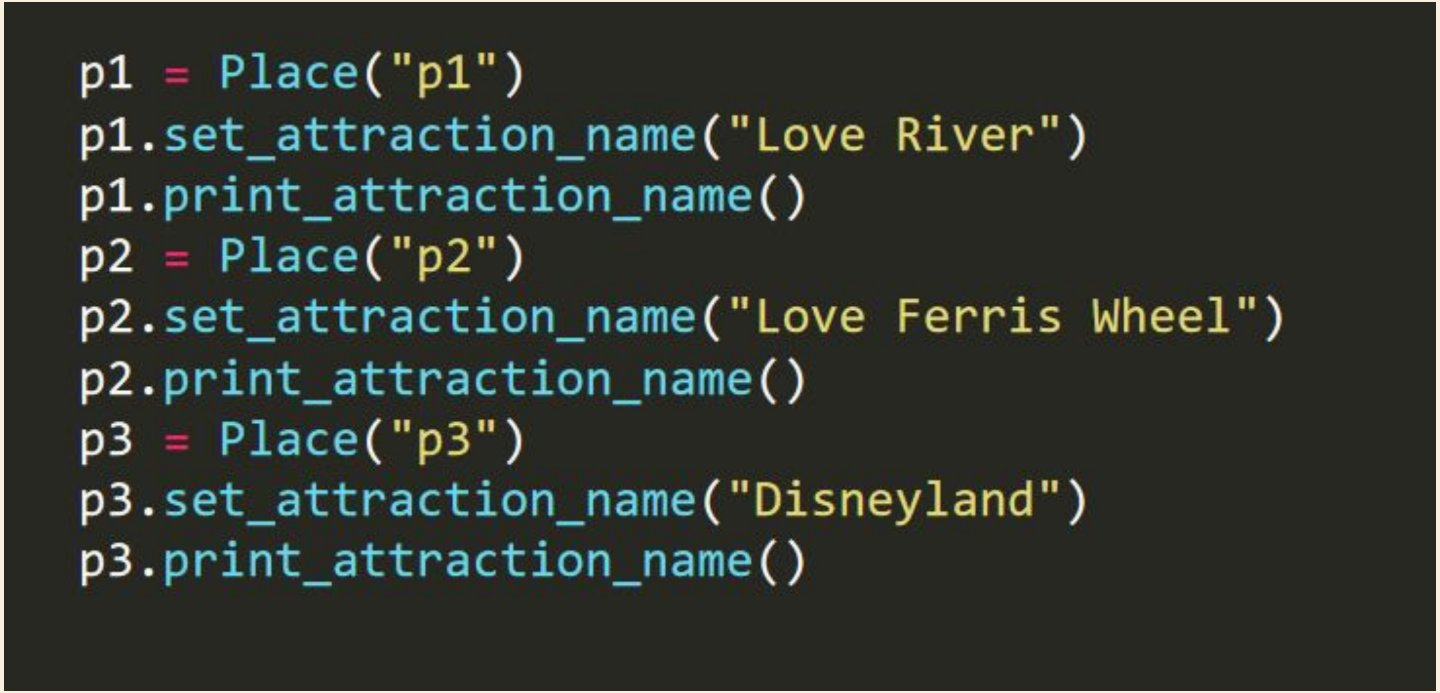
subclass **Place** must be defined as



```
class Place(Attraction):

    def __init__(self,name):
        #implement by yourself

    def set_attraction_name(self,attraction_name):
        #implement by yourself

    def print_attraction_name(self):
        #implement by yourself
        #You need to use the method "print()" defined in class "Attraction" here.
```

EXERCISE 2

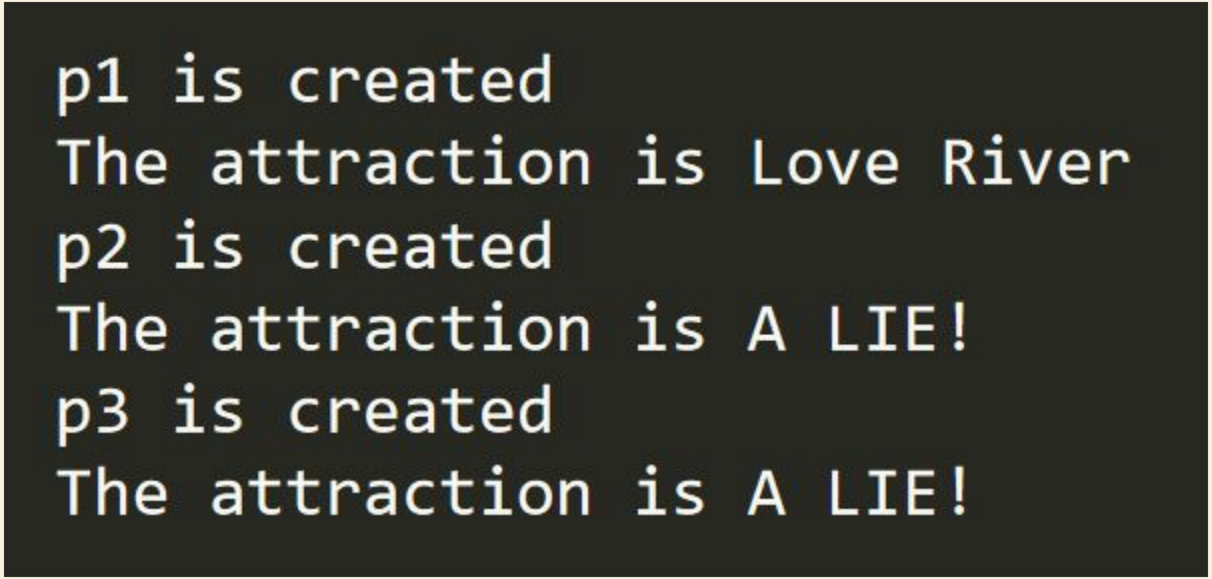
The code in your main must be like



```
p1 = Place("p1")
p1.set_attraction_name("Love River")
p1.print_attraction_name()
p2 = Place("p2")
p2.set_attraction_name("Love Ferris Wheel")
p2.print_attraction_name()
p3 = Place("p3")
p3.set_attraction_name("Disneyland")
p3.print_attraction_name()
```

EXERCISE 2

The output strings must be like



```
p1 is created
The attraction is Love River
p2 is created
The attraction is A LIE!
p3 is created
The attraction is A LIE!
```