OBJECT ORIENTED PROGRAMMING WITH PYTHON

Second Class

1st Semester

Importing a Single Class

Importing Multiple Classes from a Module

Importing an entire module

The *Python standard library*

Reading from a file

Writing to a File

Appending to a File

IMPORTING A SINGLE CLASS

- Let's create a module containing just the **Restaurant** class.
- Name the file ***restaurant.py***
- *The file contents :*

```
class Restaurant:
    def __init__(self, restaurant_name, cuisine_type):
        self.restaurant_name = restaurant_name
        self.cuisine_type = cuisine_type

    def discribe_restaurant(self):
        print(f""Welcome to {self.restaurant_name} Restaurant and the delicious dishes frome the {self.cuisine_type} Cuisine .."")

    def open_restaurant(self):
        print(" our Restaurant is open Now ^_^ ")
```

IMPORTING A SINGLE CLASS

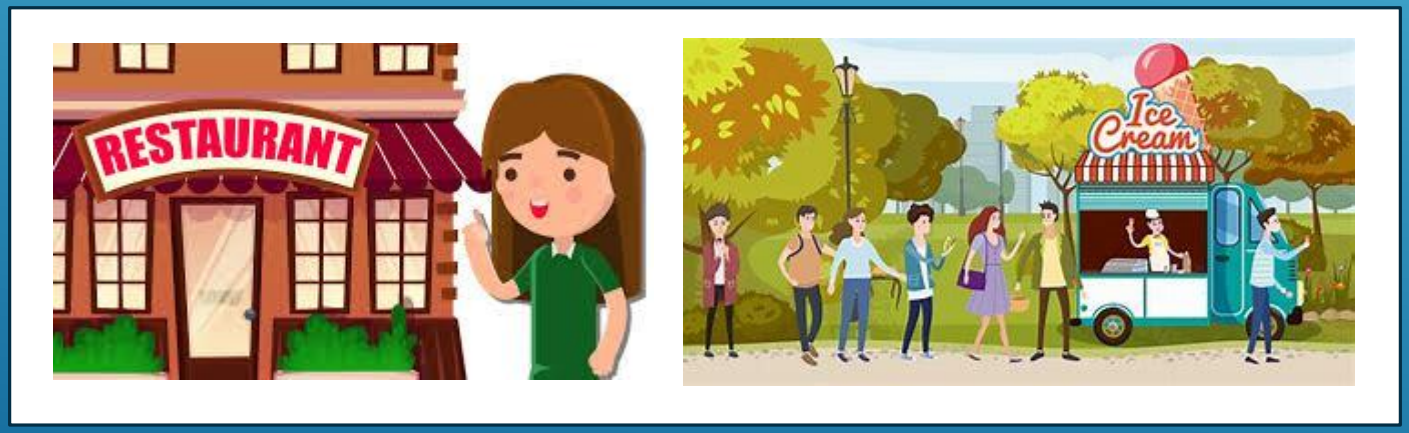
In a new python file we will import the Restaurant class as follows

```
from restaurant import Restaurant

asian_rest = Restaurant('Shanghai' , 'Asian')
asian_rest.discribe_restaurant()
asian_rest.open_restaurant()
euro_rest =Restaurant ('Big Pizza' , 'Italian')
euro_rest.discribe_restaurant()
euro_rest.open_restaurant()
kfc_rest=Restaurant('KFC' ,'Fast Food')
kfc_rest.discribe_restaurant()
```

IMPORTING MULTIPLE CLASSES FROM A MODULE

- You can store as many classes as you need in a single module, **although** each class in a module should be related somehow.
- The classes Restaurant and IceCreanStand both help represent types of restaurants.



IMPORTING MULTIPLE CLASSES FROM A MODULE

```
class Restaurant:
    def __init__(self, restaurant_name, cuisine_type):
        self.restaurant_name = restaurant_name
        self.cuisine_type = cuisine_type
        self.number = 0

    def set_number_served(self, number):
        self.number = number

    def increment_number_served(self, add):
        self.number += add

    def discribe_restaurant(self):
        print(f""Welcome to {self.restaurant_name} Restaurant and the delicious dishes frome the {self.cuisine_type} Cuisine .."")

    def open_restaurant(self):
        print(" our Restaurant is open Now ^_^ ")

class IceCreamStand(Restaurant):
    def __init__(self, restaurant_name,cuisine_type):
        super().__init__(restaurant_name,cuisine_type)
        self.flavors=['strawberry','peach','apple','watermelon']
        self.stores='MADO'

    def display_flav(self):
        print(f" hi our {self.stores} store has four flavours:")
        print(self.flavors)
```

IMPORTING MULTIPLE CLASSES FROM A MODULE

You can import as many classes as you need into a program file. If we want to make a regular **restaurant** and an **ice cream stand** in the same file, we need to import both classes, Restaurant and **IceCreamStand** :

```
from restaurant import Restaurant, IceCreamStand


kfc_rest=Restaurant('KFC' ,'Fast Food')
kfc_rest.discribe_restaurant()

icecreram=IceCreamStand('HappyDay','IceCream')
icecreram.discribe_restaurant()
icecreram.display_flav()
```

IMPORTING AN ENTIRE MODULE

You can also import an entire module and then access the classes you need using dot notation. This approach is simple and results in code that is easy to read.

```
import RestaurantClass

kfc_rest=RestaurantClass.Restaurant('KFC' ,'Fast Food')
kfc_rest.discribe_restaurant()

icecreram=RestaurantClass.IceCreamStand('HappyDay','IceCream')
icecreram.discribe_restaurant()
icecreram.display_flav()
```

THE *PYTHON STANDARD LIBRARY*

- The *Python standard library* is a set of modules included with every Python installation.
- You can use any function or class in the standard library by including a simple import statement at the top of your file.
- **random**, module can be useful in modeling many real-world situations,
- One interesting function is **randint().** This function takes two integer arguments and returns a randomly selected integer between (and including) those numbers.

```
from random import randint
RandNo=randint(1, 6)
print(RandNo)
```

Another useful function is **choice()**. This function takes in a list or tuple and returns a randomly chosen element:

```
from random import choice
players = [‘Ahmad', ‘Salim', ‘Rana', ‘Reem', ‘Ali']
first_up = choice(players)
print( first_up)
```

READING FROM A FILE

- To begin, we need a file with a few lines of text in it.
- create one (from **file menue** → New → File), name it (**text.txt**) fill it with text

```
with open('text.txt') as file_object:
    contents = file_object.read()
print(contents)
```

- The **open()** function needs one argument: the name of the file you want to open
- Python looks for this file in the directory where the program that's currently being executed is stored.
- returns an object representing *text.txt*. Python assigns this object to **file_object**, which we'll work with later in the program.
- The keyword **with** closes the file once access to it is no longer needed.
- If the text file located in another directory we can add the file path as follows:

```
with open('F:\\OOP with Python2020\\text.txt') as file_object:
    contents = file_object.read()
print(contents)
```

WRITING TO A FILE

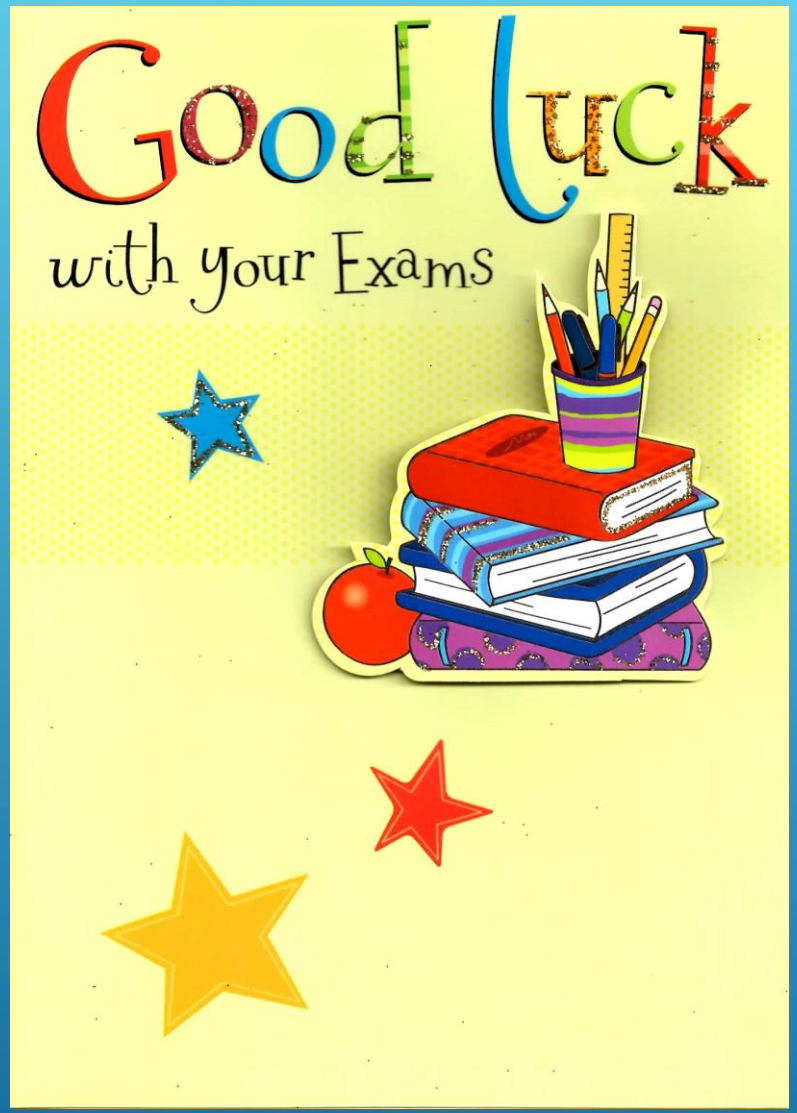
- To write text to a file, you need to call open() with a second argument telling Python that you want to write to the file

```
filename = 'programming.txt'
with open(filename, 'w') as file_object:
    file_object.write("I love programming.")
```

- ***'w'***, tells Python that we want to open the file in *write mode*.
- **Note:** You can open a file in *read mode* ('r'), *write mode* ('w'), *append mode* ('a') Or Python opens the file in read-only mode by default.
- write() method on the file object to write a string to the file.

```
filename = 'programming.txt'
with open(filename, 'w') as file_object:
    file_object.write("I love programming.\n")
    file_object.write("I love creating new games.\n")
```

```
#Write A TEXT File
filename = 'programming.txt'
with open(filename, 'w') as file_object:
    file_object.write("I love programming.\n")
    file_object.write("I love creating new games.\n")
#Appending A File
with open(filename, 'a') as file_object:
    file_object.write("I also love finding meaning in large datasets.\n")
    file_object.write("I love creating apps that can run in a browser.\n")
#Writing File Contents
with open(filename) as file_object:
    contents = file_object.read()
    print(contents)
```


Good luck
with your Exams