Data Science & Artificial Intelligence

Python For Data Science

Classes & Modules

Lecture No.- 01

By- Satya sir

Recap of Previous Lecture

- Set methods | Operations
- Dictionaries
- Functions
  - Recursion

Topics to be Covered

- File Handling in Python

- OOPs Concepts

Topic : File Handling & OOPs Concepts

File Handling

- File : The collection of related data

- Types of files : Text File (Notepad .txt), Songs | audio files (.mp3, .wav, .m4a ...), Image File (.jpg, .bmp, .gif ...), Source files (.c, .cpp, .java, .py, .js ....), Documents (.doc, .ppt, .xls, .pdf ....) ...

- Input can be given either at Compile time (Static input) or at Runtime (Dynamic input)
  - Entered by user while Execution : i/p ← 
  - may be feeded from File Directly : i/p ← File ← 

Topic : File Handling & OOPs Concepts

The Steps for file Handling

1) Open File — Open ('File Name', mode) Ex: open ('abc.txt', 'r')

2) Read/write from/into the file

3) Close File

abc.txt

Welcome Everyone

File Handling in Python is Easy

Read from file

```
with open('abc.txt', 'r') as File :
    x = File.read()  (or) File.readLine()
    Print(x, end=';')
```

(OR)

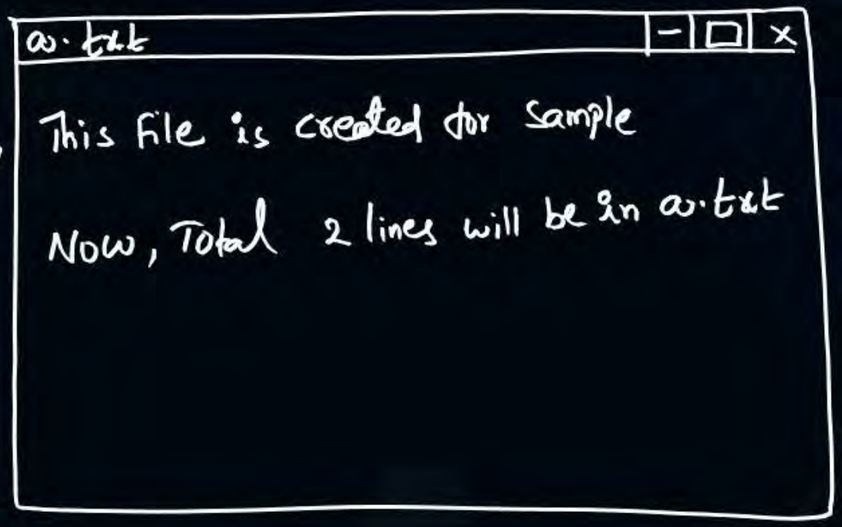
```
f = open('abc.txt', 'r')
for i in f :
    Print(i, end=';')
```

o/p: Welcome Everyone ; File Handling in Python is Easy ;

Topic : File Handling & OOPs Concepts

- Let a.txt be an Empty file

```
f = open('a.txt', 'w')
f.write('This File is created for sample')
f.write('Now, Total 2 lines will be in a.txt')
```



Close File

```
object.close()
```

Ex:

```
f = open('a.txt', 'r')
for each in f:
    Print(each)
f.close()
```

Topic : File Handling & OOPs Concepts

Modes:

r: open an existing file for a read operation.

w: open an existing file for a write operation. If the file already contains some data, then it will be overridden but if the file is not present then it creates the file as well.

a: open an existing file for append operation. It won't override existing data.

r+: To read and write data into the file. This mode does not override the existing data, but you can modify the data starting from the beginning of the file.

w+: To write and read data. It overwrites the previous file if one exists, it will truncate the file to zero length or create a file if it does not exist.

a+: To append and read data from the file. It won't override existing data.

Topic : File Handling & OOPs Concepts

```
import os

def create_file(filename):
    try:
        with open(filename, 'w') as f:
            f.write('Hello, world!\n')
        print("File " + filename + " created successfully.")
    except IOError:
        print("Error: could not create file " + filename)

def read_file(filename):
    try:
        with open(filename, 'r') as f:
            contents = f.read()
            print(contents)
    except IOError:
        print("Error: could not read file " + filename)
```

```
def append_file(filename, text):
    try:
        with open(filename, 'a') as f:
            f.write(text)
        print("Text appended to file " + filename + " successfully.")
    except IOError:
        print("Error: could not append to file " + filename)

def rename_file(filename, new_filename):
    try:
        os.rename(filename, new_filename)
        print("File " + filename + " renamed to " + new_filename + "
successfully.")
    except IOError:
        print("Error: could not rename file " + filename)
```

Topic : File Handling & OOPs Concepts

```
def delete_file(filename):
    try:
        os.remove(filename)
        print("File " + filename + " deleted successfully.")
    except IOError:
        print("Error: could not delete file " + filename)


if __name__ == '__main__':
    filename = "example.txt"
    new_filename = "new_example.txt"

    create_file(filename)
    read_file(filename)
    append_file(filename, "This is some additional text.\n")
    read_file(filename)
    rename_file(filename, new_filename)
    read_file(new_filename)
    delete_file(new_filename)
```

Topic : File Handling & OOPs Concepts

OOPs Concepts : Object-Oriented Programming : Approach of Programming to handle (or) access data through objects.

Object : Instance (or) Variable (or) Physical form of a class.

Class : It is blue Print (or) template of an object. In Python Each Programming Element is a class.

Pre-defined classes : int, float, str, bool, complex, None, list, set, Tuple, dict, Function...

Ex:
```
a=4 # a is an object of type<class 'int'>
b='GATE' # b is an object of type <class 'str'>
c=[10,'A',4.27,False] # c is an object of type <class 'list'>
```

Topic : File Handling & OOPs Concepts

OOPs concepts:

- class — blue-print for an object
- object — Instance of a class
- Inheritance — Process of acquiring, Properties of one class by another class.
- Polymorphism — The ability to use one in many forms
- Encapsulation — Wrapping up of data and code together
- Abstraction — Hiding implementation Details.

Topic : File Handling & OOPs Concepts

Inheritance

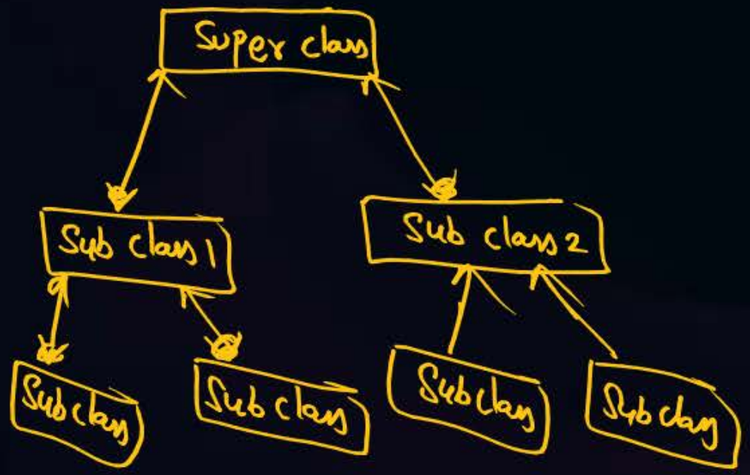
- 5 Types of Inheritance 1) Single-level 2) Multilevel 3) Hierarchical 4) Multiple 5) Hybrid.

- The class, that got inherited by Other class : Super class (or) Base class (or) Parent class.

- The class, that inherits other class : Sub class (or) Derived class (or) Child class.

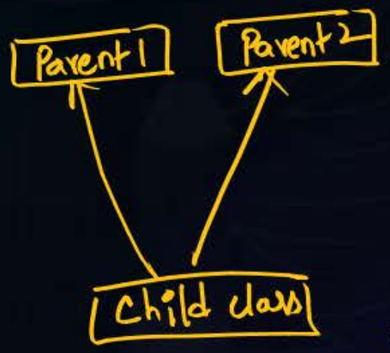
Single level

Hierarchical

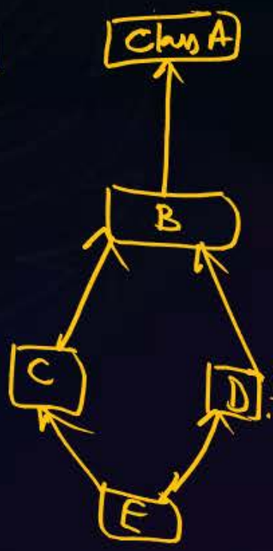


Multilevel

Multiple



Hybrid



Topic : File Handling & OOPs Concepts

#default __init__() method:

```
def __init__(self):
    Pass
```

Example :

```
class A :
    z=0
    def __init__(self, x, y):
        self.x = x
        self.y = y
        A.z = self.x + self.y
    def display( ):
        Print('The result is', A.z)
ob= A(4, 5)
ob.display( )  #9
```

Single-level Inheritance

```
class A :    # Super class
    def display( ):
        Print('I am class A')

class B ( A ) : # Sub class
    def show( ):
        Print('I am class B')

ob= B( )
ob. display( ) # I am class A
ob. show( )  # I am class B
```

Multiple Inheritance

```
class A :
    :
    :
class B:
    |
    |
class C(A, B) :
    :
    :
```

```
ob1= A( )
ob1. display( ) ✓
ob1. show( )#Invalid, Error
```

2 mins Summary

- File Handling
  - Open file
  - Read, write, append
  - Close file
- OOPs Concepts

To be Contd...

THANK - YOU