Objects & Classes

Class, methods and Object

```
Class classname:

    def __init__(self):

        self.val1 := define

        self.val2:= define

    def display():

        print(val1,val2)      #define method

 ob1=classname(val1,val2)

print(ob1)
```

Example

```python
"""Class Objects and Functions"""
class Person:
    def __init__(self, n, a):
        self.name = n
        self.age = a

    def setNewName(self, n1):
        self.name = n1

    def setNewAge(self, a1):
        self.age = a1

    def displayValues(self):
        print("Name: {} and Age : {}".format(self.name, self.age))
ob = Person("John", 21)
ob.displayValues()
ob.setNewName("Harry")
print("After setting new name")
ob.displayValues()
ob.setNewAge(23)
print("After setting new age")
ob.displayValues()
```

Result:

```
Name: John and Age : 21
After setting new name
Name: Harry and Age : 21
After setting new age
Name: Harry and Age : 23
```

Polymorphism

```python
"""Polymorphism"""
class duck:
    def __init__(self):
        print("I am a Duck first")
    def mySound(self):
        print("Quack Quack")
class crow:
    def __init__(self):
        print("I am a hen now")
    def mySound(self):
        print("caw caw")


ob1=duck()
ob1.mySound()
ob2=crow()
ob2.mySound()
```

Result

```
I am a Duck first
Quack Quack
I am a hen now
caw caw
```

Inheritance

```python
"""Inheritance"""
class parent:
    def __int__(self):
        print("Hello Parent class")
    def parentPrint(self):
        print("Hello parent class")


class child(parent):
    def __int(self):
        print("I am in child class now")
    def childPrint(self):
        print("Hello child class")


ob=child()
print("Calling parent method from child class....")
ob.parentPrint()
print("Calling child method from child class...")
ob.childPrint()
```

Result:

```
Calling parent method from child class....
Hello parent class
Calling child method from child class...
Hello child class
```