

Object Oriented Programming Concept

```
In [ ]: Object Oriented Programming --> It is just a programming concept in which each and every program will be implemented
        with the help of classes and objects.

Functional programming --> It is also a programming concept in which each and every program is implemented with the help of
        Functions.

Object Oriented Programming --> Java, C++, C Sharp, Python and Javascript

Object --> Any real world entity is an object.
        --> Each and Every object is Having behaviour and Properties.

Example of Behaviour and Properties of an Object:

    Student(Object) --> Behaviour --> Learning,Writing,Thinking,Reading
                    --> Properties --> Name,Class,Roll,Grade,Height,Weight

    Mobile(Object)  --> Behaviour --> Calling, Texting,Capture_Camera,Speaker,Bluetooth,Wifi
                    --> Properties --> Color,Ram,operating System,DisplaySize,Memory

    Laptop(Object)  --> Behaviour --> Webcam,Gaming, IDES,ZOOMING,VideoGraphy, Browsing, PAINT,MSWORD,MSEXCEL
                    --> Properties --> COLOR,BRAND, RAM,ROM,SSD,GRAPHIC CARD,DISPLAY

    Human(Object)   --> Behaviour --> Talking,Thinking,Learning,Dancing,Singing,Sleeping etc
                    --> Properties --> Appearance,Name,height,Weight,BMI

    Dog(Object)      --> Behaviour --> Barking,Running,Garding
                    --> Properties --> Color,Breed,Legs,Hair
```

OOPS VS Functional Programming

```
In [ ]: Object oriented programming is much secure as compare to functional programming
        because in object oriented programming we can hide our data/variables.
        In object orinted programming we are binding the variables and methods together and we can
        use them by creating the object
```

Class

```
In [ ]: Class --> Class is a blueprint/Plan of an object.
        --> Classes are basically use to represent behaviour(methods) and properties
            (variables) of an object
```

Example:

```
class Television:
    def __init__(self):
        self.size=10
        self.color=Blue
        self.brand=LG
        self.Model=11th Gen
    def channel(self):
        pass
    def Volume(self):
        pass
    def bluetooth(self):
        pass
    def wifi(self):
        pass
    def mobile_connect(self):
        pass
```

```
In [ ]: class House:
    def __init__(self):
        self.Color=Green
        self.city=LK0

    def dinig_room(self):
        pass
    def kitchen(self):
        pass

    def hall(self):
        pass
```

Object

```
In [ ]: Object --> Physical Existance of a Class
        --> Any real world entity is an object.
```

We can create any number of object **for** a **class**

Syntax of Creating a Class

```
In [ ]: Syntax of Creating a Class:
        class class_name:
```

Syntax for creating an object

```
In [ ]: Syntax for creating an object:
        reference_variable = class_name()
```

Reference Variable

```
In [ ]: Reference Variables --> Is used to access the properties and behaviour of an object.Basically it is
        holding the memeory address of an object.
        --> In python if you are not giving any reference variable then python will have garbage collector
        that will delete useless object.
        --> if you are not giving reference variable then that object is useless as we cannot
        access the properties and behaviour of a object
```

Example to Demonstrate Classes , Object and Reference Variable

```
In [7]: #demo
class Student:
    def __init__(self,name,rollno,marks):
        self.name=name
        self.rollno=rollno
        self.marks=marks

    def Student_detail(self):
        print("Student name is :",self.name)
        print("Student rollno is :",self.rollno)
        print("Student marks is :",self.marks)

x = Student("Akash", "99", "99")
x.Student_detail()
```

Student name is : Akash
Student rollno is : 99
Student marks is : 99

Constructor in Python

```
In [ ]: Constructor --> It is special Method
        --> In python constructor name always be __init__.
        --> Whenever an object creation will be done Python will automatically execute the
            Constructor.
        --> If you are not giving your own constructor then PVM will automatically execute
            its own default constructor
        --> For every object constructor will executed only once.
        --> Constructor will take atleast one argument and that is self.
```

Example to Demonstrate Constructor

```
In [19]: class Animal:
    def __init__(x):
        print("Constructor is executed")
    def f1(x):
        print("f1-Method Exceuted")

x=Animal()
y=Animal()
z=Animal()
```

Constructor is executed
Constructor is executed
Constructor is executed

self variable

```
In [ ]: self --> self is a variable that is used to point to the current object.
        --> if you are not using self variable then whatever first variable you are going to
            pass in your constructor that will be internally considered as a self.
```

Example to Demonstrate Constructor and Self Variable

```
In [24]: #demo
class Student:
    def __init__(self,name,rollno,marks):
        self.name=name
        self.rollno=rollno
        self.marks=marks
        print("Student name is :",self.name)
        print("Student rollno is :",self.rollno)
        print("Student marks is :",self.marks)

x = Student("Akash", "99", "99")
print("-----")
print("-----")
y=Student("Arnav", "100", "200")
print("-----")
```

Student name is : Akash
Student rollno is : 99
Student marks is : 99

Student name is : Arnav
Student rollno is : 100
Student marks is : 200

```
In [25]: #demo
class Student:
    def __init__(arnav,name,rollno,marks):
        arnav.name=name
        arnav.rollno=rollno
        arnav.marks=marks
        print("Student name is :",arnav.name)
        print("Student rollno is :",arnav.rollno)
        print("Student marks is :",arnav.marks)

x = Student("Akash", "99", "99")
print("-----")
print("-----")
y=Student("Arnav", "100", "200")
print("-----")
```

Student name is : Akash
Student rollno is : 99
Student marks is : 99

Student name is : Arnav
Student rollno is : 100
Student marks is : 200
