**CLASS and OBJECT**

* Classes are defined to implement oops(object oriented programming)
* It consist of two things
  + classes
  + object
  + syntax:
    - class <classname>:
      * --
      * --
      * --
  + class has mainly two things.
    - Attribute:what
    - Function: on what the functionality should be appiled
* We can also declare an empty class but when it is executed it gives error so to avoid that error we use **pass**
* The variables declared in the class are called class variables
* If the method declared inside the class are called regular method
* **Object:**
  + Object is created for the class and it is used to call or access the variables and methods in the class
  + We can access the methods using the attribute fetching operator(.) i.e dot operator
  + **Syntax**: objectname=classname()
  + object is also known as “instance”
* **References:**
  + Classes and objects uses the two types of references they are
    - 1.object reference
    - 2.class reference
* **Methods :**
  + There are three methods used in the classes they are
    - Regular method
    - Class method
    - Static method
* When two different objects call the same function it get into to ambiguity so we use another parameter along with the parameters that are used for the function is “**self”**
* **Def add (self,a,b)**
  + **Self.a=a**
  + **Self.b=b**
  + **Self.sum=self.a+self.b**
  + Whatever the method uses the “self” as a first parameter it is known as regular method
  + All the regular methods are accessible by only the objects but not classes
  + **Note:** python does not have the getter() and the setter() methods
* **Constructor and destructor:**
  + Constructor is also a method which having the same class name
  + Whenever we create a object the constructor is automatically called ,we don’t need to access them using the object
  + **Syntax:** 
    - **objectname=classname()**
  + This object hits a method called as constructor that is \_\_init\_\_()
  + Constructors maybe default or parameterised
  + ex:
    - class Arith()
      * def \_\_init\_\_(self)default
      * def\_\_init\_\_(self,name)
  + print(”haii”+name)
    - o=Arith(xyz)
  + **Result:**
    - haii xyz
* **Destructor:**
  + it is used to delete the constructor
  + we have to call it manually by \_\_del\_\_()