**OOPS**

OOP :

OOP stands for Object oriented programming.

Real-time entities called objects.

It is used for large & complex projects.

Class : It acts as blue print / plan/model/design for objects

Object : A physical existence of a class.

We can create any number of objects after creating a class.

In Object oriented programming class name should start with upper case.

Every object has properties and behaviour. Properties(data) are specified by variables and behaviours can be specified by methods.

Variables :

There are 3 types of variables -

1. Instance variable (Object level variable)
2. Static variable (Class level variable)
3. Local variable (Method level variable)

Methods :

There are 3 types of methods –

1. Instance Method
2. Class Method
3. Static Method

Reference variable :

Reference variable can be used to refer objects, by using that variable we can invoke the required functionality of object.

For single object there may be a chance of multiple references.

Eg :

reference\_name = ClassName()

class Student:  
 *"This is document string of a class"*print(Student.\_\_doc\_\_)

class Student:  
 def \_\_init\_\_(self):  
 self.name = 'Sai'  
 self.rollno = 118  
 self.marks = 98  
  
 def talk(self):  
 s = 10  
 print('I am :', self.name)  
 print('Roll no :', self.rollno)  
 print('Marks : ', self.marks)  
  
s1 = Student()  
s2 = Student()  
print(s1.name)  
print(s1.rollno)  
print(s1.marks)  
print(s1.talk())  
  
print(s2.name)  
print(s2.rollno)  
print(s2.marks)  
print(s2.talk())  
  
print(id(s1)) #2036884471760  
print(id(s2)) #2036884471760  
  
print(s1.name)  
print(s2.name)

Whenever we create an object constructor will get executed automatically and provides values to the attributes.

For every object constructor will get executed separately. For same class the data may be same but objects will have different addresses.

Self Variable

Self is a reference variable which is pointing to current object always.

Inside a class we require ‘self’ to access the variables but outside the class we require reference variable to access the variables because inside class reference variables (s1, s2) are not present but self is there.

The first argument to the constructor is always ‘self’.

The first argument to the instance method is always “self”.

We are not required to provide value for self variable. PVM itself will provide value.

We can use self always inside class to declare object related variables.

Self is not a keyword. We can use any name like delf, kelf etc.