- **Class and Object**

- Class is the description of an object.
- Class is the blueprint of an object.
- Object is the real world entity which have state (properties) and behaviour.
- Example of object car, pen, student, employee etc.

- **Defining class in Python:**

- To define a class we have to use the keyword class.
- Syntax:

```
class <NameofClass>:
    // static variables (class level variables)
    #methods
    def __init__(self,paramters..):
        // instance variables

    def <NameofMethod>(self,parameters..)
        // methods accessing instance variables
    @staticMethods
    def <NameofMethod>(parameters..):
        //utitlity methods
    @classMethods
    def <NameofMethod>(cls,parameters)
        // accessing class level variables
```

Code Example

```python
class Student:
    college_name='softwarica college'
    def __init__(self,name,rollno,address):
        self.name=name
        self.rollno=rollno
        self.address=address
    def display(self):
        print('Name:',self.name)
        print('Roll No:',self.rollno)
        print('Address:',self.address)
        print('College Name:',self.college_name)
    @staticmethod
    def fee(self,adm_fee,sem_fee):
        total_fee=adm_fee+sem_fee
        return total_fee
    @classmethod
    def change_col_name(cls,newname):
        cls.college_name=newname
```

- **Special method in Python:**

  def _ _init_ _(self):
  //instance variables
- Automatically called when object created.
- To declare and initialize instance variable only once.
- At least one argument self.
- Python will provide default _ _init_ _ method if we do not provide.

- **Self**

- self variable is a reference variable which always pointing to current object.

- In Python self refer to the current object.

- The first argument to the constructor and instance method should be self.

- PVM is responsible to provide value for self argument and we are not required to provide explicitly.

- By using self we can declare instance variables with the class.

- By using self we can access instance variables within class.

Note: In case of self we can write any name.it is user-define name.

- **Inside Python class:-**

- Variables: Types of variables

    1. Instance variables

    2. Static variables (class variables)

    3. Local variables

- Methods: Types of Methods

    1. Instance Methods

    2. Class Methods

    3. Static Methods

- **Instance Variable: object level variables**

- Variable which are declared with the help of self.

- vary to object to object

- Separate copy of instance variable is created for every object.

Example:

```
 1    class Student:
 2        def __init__(self, x, y):
 3            self.name = x
 4            self.rollno = y
 5
 6        def display(self):
 7            print('hello i m', self.name)
 8            print('my roll no is', self.rollno)
 9
10
11    s1 = Student('shyam', 100)
```

Note: Suppose all object have same college name, why to have same in all object? so in-order to save the space we make the class level variable known as static variable.

- **static variable: class level variable**

- Only one copy of variable is created and shared by every object.

- **local variable: Method level variable**

- Variable declared inside method for temporary purpose.

......coming up Methods........