Agenda :

1) oops
2) Create class & objects
3) Access modifier.. + Constructor.

Intro to oops:

Programming Paradigms: 1) Procedural
2) oop

Procedural Program: 1) Set of instructions
2) functions / methods

main.Py

```
def  Print_hello():
            Print("hello")


def   Print_age(age):
          Print(age)
Printhello()
Print age()
```

shoP.Py

```
Sales_1  =  3000              # Ashutosh_ Shop
  marketing  =  1000
   raw_material  =  500


  def Profit(Sales, marketing, raw_material):
                  sales - marketing - raw-material;
```

Sales_2 = 5000 # Vijya_Shop

marketing_1= 2000

raw_material_1= 1500

Karan is teaching LLD

We are Studing LLD

India won T20

oops is intresting

Subject + verb

Print_user_data(user_id, user_name, PSP)..

doing something

Student someone

oops ⟶ entities : real world objects

Student

- psp
- email
- batch

has ⇒ attributes

- attendence
- attend_class()
- solve_assign
- give_mock()

does ⇒ methods

Rajiv = Student ( )

Rajiv. psp

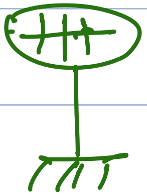
1 Principle with 3 Pillars:

foundation/concept   support/hold together

Principle : Abstraction

represent things as an idea

entity.

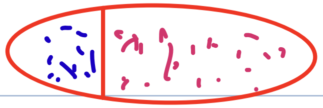


.Sort( )

Pillar :

i) encapsulation

capsule



1) Hold medicine together

2) Protect medicine from outer world.

# 1) binding attributes & methods

2) Protect attributes & behaviour.

Access modifi

Access modifiers

1) Public # Bus

2) Protected # Car

3) Private. # mobile

```
class Person:
        def __init__(self):
                self.name = "Karan".        #Public


class Z:
        p = Person()
        Print (p.name)


class PersonPrivate:
        def __init__(self):
                self.__name = "Karan"


class Protectedclass:
        def __init__(self):
```

```
self._name = "Kazan"
```