In [1]:

```
#classes and object
```

In [8]:

```
class Car:
    x='My Car Name is Creta'

c=Car()
print(c.x)
```

```
My Car Name is Creta
```

In [9]:

```
#__init__
```

In [17]:

```
class car:
    def __init__(self,name,price,color):
        self.n=name
        self.p=price
        self.c=color


c=car('Creta','17lakh','Black')
print('Name of car:',c.n)
print('Price of car:',c.p)
print('Color of car:',c.c)

c1=car('Amaze','20lakh','white')
print('Name of car:',c1.n)
print('Price of car:',c1.p)
print('Color of car:',c1.c)
```

```
Name of car: Creta
Price of car: 17lakh
Color of car: Black
Name of car: Amaze
Price of car: 20lakh
Color of car: white
```

In [23]:

```
class car:
    def __init__(self,name,price,color):
        self.n=name
        self.p=price
        self.c=color

    def show_data(self):  #method
        print('Name of car:',self.n)
        print('Price of car:',self.p)
        print('Color of car:',self.c)

c1=car('Swift','14lakh','red')
c1.show_data()
c2=car('Wagnor','9lakh','gray')
c2.show_data()
```

```
Name of car: Swift
Price of car: 14lakh
Color of car: red
Name of car: Wagnor
Price of car: 9lakh
Color of car: gray
```

In [30]:

```
class student_details:
    def __init__(swati,name,rollno,course,fees):
        swati.name=name
        swati.rn=rollno
        swati.c=course
        swati.f=fees


    def display(swati):
        print('Name:',swati.name)
        print('Rollno:',swati.rn)
        print('Course name:',swati.c)
        print('Fees of course:',swati.f)


riya=student_details('Riya Sharma',154,'Python',45000)
rohan=student_details('Rohan Verma',155,'Data Analytics',60000)
kiran=student_details('Kiran Chaudhari',145,'SAP',56789)
neha=student_details('Neha Rao',123,'software Testing',456789)
```

In [32]:

```
riya.display() #object_name.function_name
```

```
Name: Riya Sharma
Rollno: 154
Course name: Python
Fees of course: 45000
```

In [33]:

```
kiran.display()
```

```
Name: Kiran Chaudhari
Rollno: 145
Course name: SAP
Fees of course: 56789
```

In [34]:

```
print(neha.c)  #object_name.attribute_name
```

```
software Testing
```

In [40]:

```
class student_details:
    def __init__(swati,name,rollno,course,fees):
        swati.name=name
        swati.rn=rollno
        swati.c=course
        swati.f=fees


    def display(swati):
        print('Name:',swati.name)
        print('Rollno:',swati.rn)
        print('Course name:',swati.c)
        print('Fees of course:',swati.f)


riya=student_details('Riya Sharma',154,'Python',45000)
rohan=student_details('Rohan Verma',155,'Data Analytics',60000)
kiran=student_details('Kiran Chaudhari',145,'SAP',56789)
neha=student_details('Neha Rao',123,'software Testing',456789)


kiran.c='Data Science' #modify
kiran.f=65000
riya.name='Riya Joshi'

kiran.display()
riya.display()
```

```
Name: Kiran Chaudhari
Rollno: 145
Course name: Data Science
Fees of course: 65000
Name: Riya Joshi
Rollno: 154
Course name: Python
Fees of course: 45000
```

In [46]:

```
class student_details:
    def __init__(swati,name,rollno,course,fees):
        swati.name=name
        swati.rn=rollno
        swati.c=course
        swati.f=fees


    def display(swati):
        print('Name:',swati.name)
        print('Rollno:',swati.rn)
        print('Course name:',swati.c)
        print('Fees of course:',swati.f)


riya=student_details('Riya Sharma',154,'Python',45000)
rohan=student_details('Rohan Verma',155,'Data Analytics',60000)
kiran=student_details('Kiran Chaudhari',145,'SAP',56789)
neha=student_details('Neha Rao',123,'software Testing',456789)


del riya.c

print(riya.name)
print(riya.rn)
print(riya.f)
print(riya.c)
```

```
Riya Sharma
154
45000

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3680\1968641516.py in <module>
     25 print(riya.rn)
     26 print(riya.f)
---> 27 print(riya.c)

AttributeError: 'student_details' object has no attribute 'c'
```

In [48]:

```
class student_details:
    def __init__(swati,name,rollno,course,fees):
        swati.name=name
        swati.rn=rollno
        swati.c=course
        swati.f=fees


    def display(swati):
        print('Name:',swati.name)
        print('Rollno:',swati.rn)
        print('Course name:',swati.c)
        print('Fees of course:',swati.f)


riya=student_details('Riya Sharma',154,'Python',45000)
rohan=student_details('Rohan Verma',155,'Data Analytics',60000)
kiran=student_details('Kiran Chaudhari',145,'SAP',56789)
neha=student_details('Neha Rao',123,'software Testing',456789)

del riya

riya.display()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3680\2483828759.py in <module>
     21 del riya
     22 
---> 23 riya.display()

NameError: name 'riya' is not defined
```

In [50]:

```
class msg:
    def __init__(self):
        print('GOOD MORNING EVERYONE')


m1=msg()
```

```
GOOD MORNING EVERYONE
```

```
#classs and objects: - object oriented programming = OOP

4 pillars of OOPs: -

1. Inheritance
2. Polymorphism
3. Encapsulation
4. Abstraction
```

In [53]:

```
class emp:
    def __init__(self,name,id,designation):
        self.n=name
        self.id=id
        self.d=designation

    def show(self):
        print("Name:",self.n)
        print('ID:',self.id)
        print('Designation:',self.d)


e1=emp('Riya',234,'Python Developer')
e1.show()
```

```
Name: Riya
ID: 234
Designation: Python Developer
```

In [63]:

```
class emp:
    def __init__(self,name,id,designation):
        self.n=name
        self.id=id
        self.d=designation

    def show(self):
        print("Name:",self.n)
        print('ID:',self.id)
        print('Designation:',self.d)


class com(emp): #com class is inheriting the emp class
    pass


c1=com('Neha',234,'Java developer')
c1.show()
```

```
Name: Neha
ID: 234
Designation: Java developer
```

In [66]:

```
class emp:
    def __init__(self,name,id,designation):
        self.n=name
        self.id=id
        self.d=designation

    def show(self):
        print("Name:",self.n)
        print('ID:',self.id)
        print('Designation:',self.d)


class com(emp):
    def __init__(self,cname,caddr,name,id,designation):
        self.cn=cname
        self.ca=caddr
        emp.__init__(self,name,id,designation) #constructor

    def display(self):
        print("Name:",self.n)
        print('ID:',self.id)
        print('Designation:',self.d)
        print('Company name:',self.cn)
        print('Company Address:',self.ca)


c1=com('Wipro','Pune','Rohan',3456,'Developer')
c1.show()
```

```
Name: Rohan
ID: 3456
Designation: Developer
```

In [67]:

```
c1.display()
```

```
Name: Rohan
ID: 3456
Designation: Developer
Company name: Wipro
Company Address: Pune
```

In [78]:

```
class student:
    def __init__(self,sname,coursename):
        self.n=sname
        self.c=coursename

    def data(self):
        print('Name of student:',self.n)
        print('Name of Course:',self.c)


class trainer(student):
    def __init__(self,tname,exp,sname,coursename):
        self.t=tname
        self.e=exp
        self.l=''
       # student.__init__(self,sname,coursename)
        super().__init__(sname,coursename)


    def details(self):
        print('Trainer name:',self.t)
        print('Experience of trainer:',self.e)
        print('Student name:',self.n)
        print('Coursename:',self.c)
        print('Location:',self.l)


t1=trainer('neha','5 years','Simran','Data analytics')
t1.l='Pune'
t1.details()
print()
t2=trainer('Priya','8 years','Rahul','SAP')
t2.l='Mumbai'
t2.details()
```

```
Trainer name: neha
Experience of trainer: 5 years
Student name: Simran
Coursename: Data analytics
Location: Pune

Trainer name: Priya
Experience of trainer: 8 years
Student name: Rahul
Coursename: SAP
Location: Mumbai
```

In [71]:

```
t1.data()
```

```
Name of student: Simran
Name of Course: Data analytics
```

```
In [ ]:
```