

```
In [9]: ► class Person:
        def __init__(self,name,age):
            self.name=name
            self.age=age

        def greet(self):
            print(f'Hello Mr/Ms {self.name}. How are you? You will be {self.age} next years old')

        def speak(self):
            print('HI Idiot')
            print('How are you?')
p1=Person('Gagana',19)
p1.greet()
p1.speak()
p2=Person('Varsha',20)
p2.greet()
p2.speak()
#ClassName(Some Parameter)
```

Hello Mr/Ms Gagana. How are you? You will be 19 next years old  
 HI Idiot  
 How are you?  
 Hello Mr/Ms Varsha. How are you? You will be 20 next years old  
 HI Idiot  
 How are you?

```
In [11]: ► #Creating a student Class and multiple objects
class Student:
    def __init__(self,name,usn,branch):
        self.name=name
        self.usn=usn
        self.branch=branch
    def display_info(self):
        print(f'Name:{self.name}, USN: {self.usn}, Branch:{self.branch}')
s1=Student('Mithun','1NC22IS032','ISE')
s2=Student('Shabhas','1NC22IS050','ISE')
s1.display_info()
s2.display_info()
```

Name:Mithun, USN: 1NC22IS032, Branch:ISE  
 Name:Shabhas, USN: 1NC22IS050, Branch:ISE

In [16]:  *#Creating an Employee Class*

```
class Employee:
    def __init__(self, emp_name, emp_id, emp_sal):
        self.emp_name=emp_name
        self.emp_id=emp_id
        self.emp_sal=emp_sal #Data Members
    def read_info(self):
        self.emp_name=input("Enter the Name of the Employee:")
        self.emp_id=input("Enter the ID of the Employee:")
        self.emp_sal=input("Enter the Sal of the Employee:")
    def display_info(self):
        print(f'Employee ID :{self.emp_id}\nEmployee Name:{self.emp_name}')
e1=Employee(' ', ' ', 0.0)
e2=Employee(' ', ' ', 0.0)
e1.read_info()
e2.read_info()
e1.display_info()
e2.display_info()
```

Enter the Name of the Employee:Mithun  
Enter the ID of the Employee:ID11  
Enter the Sal of the Employee:200000  
Enter the Name of the Employee:Swaroop  
Enter the ID of the Employee:ID12  
Enter the Sal of the Employee:220000  
Employee ID :ID11  
Employee Name:Mithun  
Employee Salary:200000  
Employee ID :ID12  
Employee Name:Swaroop  
Employee Salary:220000

```
In [19]: ▶ #Creating a student Class and multiple objects
class Student:
    def __init__(self,name,usn,branch,m1,m2,m3):
        self.name=name
        self.usn=usn
        self.branch=branch
        self.m1=m1
        self.m2=m2
        self.m3=m3
    def display_info(self):
        print(f'Name:{self.name}, USN: {self.usn}, Branch:{self.branch}')
        print(f'M1:{self.m1}, \nM2:{self.m2}, \nM3:{self.m3}')
s1=Student('Mithun','1NC22IS032','ISE',20,20,19)
s2=Student('Shabhas','1NC22IS050','ISE',19,20,19)
s1.display_info()
s2.display_info()
```

```
Name:Mithun, USN: 1NC22IS032, Branch:ISE
M1:20,
M2:20,
M3:19
Name:Shabhas, USN: 1NC22IS050, Branch:ISE
M1:19,
M2:20,
M3:19
```

```
In [24]: ► class Rectangle:
    def __init__(self,l,b):
        self.l=l
        self.b=b
    def area(self):
        return self.l*self.b

l1=int(input("Enter the length of the Rectangle:"))
b1=int(input("Enter the breadth of the Rectangle:"))
r1=Rectangle(l1,b1)
print("Area of Rectangle 1 is : ",r1.area())
l2=int(input("Enter the length of the Rectangle:"))
b2=int(input("Enter the breadth of the Rectangle:"))
r2=Rectangle(l2,b2)
print("Area of Rectangle 1 is : ",r2.area())

if r1.area()>r2.area():
    print("The area of Rectangle 1 is greater than Rectange 2")
else:
    print("The area of Rectangle 1 is not greater than Rectange 2")
```

```
Enter the length of the Rectangle:2
Enter the breadth of the Rectangle:3
Area of Rectangle 1 is : 6
Enter the length of the Rectangle:4
Enter the breadth of the Rectangle:5
Area of Rectangle 1 is : 20
The area of Rectangle 1 is not greater than Rectange 2
```

```
In [13]: ► class Triangle:
    def __init__(self,b,h):
        self.b=b
        self.h=h
    def area(self):
        return self.b*self.h/2
t1=Triangle(10,15)
t2=Triangle(3,5)
t3=Triangle(4,8)

if t1.area()>t2.area() and t1.area()>t3.area():
    print("Triangle T1 contains Largest area")
elif t2.area()>t1.area() and t2.area()>t3.area():
    print("Triangle T2 contains Largest area")
else:
    print("Triangle T3 contains Largest area")
```

```
Triangle T1 contains Largest area
```

```
In [18]: ▶ #Single Inheritance
class Phone:
    def dial(self):
        print("The Phone calls the Person")

class Camera_Phone(Phone):
    def take_photo(self):
        print('It will take the photo')

pc1=Camera_Phone()
pc1.dial()
pc1.take_photo()
```

The Phone calls the Person  
It will take the photo

```
In [19]: ▶ #Multilevel Inheritance
class Phone:
    def dial(self):
        print("The Phone calls the Person")

class Camera_Phone(Phone):
    def take_photo(self):
        print('It will take the photo')

class Smart_Phone(Camera_Phone):
    def access_Internet(self):
        print('This phone gets access to Internet')

s1=Smart_Phone()
s1.dial()
s1.take_photo()
s1.access_Internet()
```

The Phone calls the Person  
It will take the photo  
This phone gets access to Internet

```
In [20]: ▶ #Multiple Inheritance
class Phone:
    def dial(self):
        print("The Phone calls the Person")

class Camera():
    def take_photo(self):
        print('It will take the photo')

class Smart_Phone(Phone, Camera):
    def access_Internet(self):
        print('This phone gets access to Internet')

s1=Smart_Phone()
s1.dial()
s1.take_photo()
s1.access_Internet()
```

The Phone calls the Person  
It will take the photo  
This phone gets access to Internet

```
In [ ]: ▶ #Heirachical Inheritance
class Camera():
    def take_photo(self):
        print('It will take the photo')

class Camera_Phone(Camera):
    def dial():
        print("The Phone calls the Person")

class Smart_Phone(Camera):
    def access_Internet(self):
        print('This phone gets access to Internet')

s1=Smart_Phone()
s1.dial()
s1.take_photo()
s1.access_Internet()
```

```
In [22]: ▶ #Hybrid Inheritance
#Multilevel Inheritance
class Phone:
    def dial(self):
        print("The Phone calls the Person")

class Camera_Phone(Phone):
    def take_photo(self):
        print('It will take the photo')

class Game:
    def play_game(self):
        print('You can play any game')

class Smart_Phone(Camera_Phone, Game):
    def access_Internet(self):
        print('This phone gets access to Internet')

s1=Smart_Phone()
s1.dial()
s1.take_photo()
s1.access_Internet()
s1.play_game()
```

```
The Phone calls the Person
It will take the photo
This phone gets access to Internet
You can play any game
```

```
In [28]: ▶ #Operator Overloading +,-,/,*,<,>

a=10
b=20
print(a+b)
print(a*b)
print(a>b)
print(a<b)
```

```
print(int.__add__(a,b))
print(int.__mul__(a,b))
print(int.__gt__(a,b))
print(int.__lt__(a,b))
```

```
30
200
False
True
30
200
False
True
```

```
In [32]: ▶ c='10'  
d='20'  
print(c+d)  
print(str.__add__(c,d))
```

1020

1020

```
In [40]: ▶ class Student:  
def __init__(self,m1,m2):  
    self.m1=m1  
    self.m2=m2  
  
def __add__(self,other):  
    tm1=self.m1+other.m1  
    tm2=self.m2+other.m2  
    print("Total marks in subject m1 is ",tm1)  
    print("Total marks in subject m2 is ",tm2)  
s1=Student(80,90)  
s2=Student(70,95)  
  
s3=s1+s2
```

Total marks in subject m1 is 150

Total marks in subject m2 is 185



```
In [53]: ▶ class Student:
    def __init__(self,m1,m2):
        self.m1=m1
        self.m2=m2

    def __add__(self,other):
        tm1=self.m1+other.m1
        tm2=self.m2+other.m2
        t=Student(tm1,tm2)
        return t

    def __str__(self):
        return f'{self.m1} {self.m2}'

s1=Student(80,90)
s2=Student(70,95)
print(s1)
print(s2)
s3=s1+s2
print(s3)

ss1=Student(8,9)
ss2=Student(7,9)
ts=ss1+ss2
print(ts)

a=10
b=20
c=a+b
print(c)
```

```
80 90
70 95
150 185
15 18
30
```

```
In [56]: ► class Student:
    def __init__(self,m1,m2):
        self.m1=m1
        self.m2=m2

    def __gt__(self,other):
        t1=self.m1+self.m2
        t2=other.m1+other.m2
        if t1>t2:
            return True
        else:
            return False

s1=Student(60,70)
s2=Student(90,85)

if s1>s2:
    print("Student 1 is the winner")
else:
    print("Student 2 is the winner")
```

Student 2 is the winner

```
In [61]: ▶ class Circle:
    def __init__(self,r):
        self.r=r

    def __gt__(self,other):
        if self.r > other.r:
            return True
        else:
            return False

    def __eq__(self,other):
        if self.r==other.r:
            return True
        else:
            return False

    def __lt__(self,other):
        if self.r < other.r:
            return True
        else:
            return False

c1=Circle(5)
c2=Circle(7)
c3=Circle(5)

print(c1>c2)
print(c1==c3)
print(c2<c3)
```

False  
True  
False

```
In [66]: ▶ class Point:
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def distance(self,other):
        dx=(other.x - self.x)**2
        dy=(other.y - self.y)**2
        a=(dx+dy)**0.5
        return a

p1=Point(1,2)
p2=Point(4,6)
d=p1.distance(p2)
print("The distance between 2 points is :",d)
```

The distance between 2 points is : 5.0

```
In [70]: ► import math
class Point:
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def distance(self,other):
        return math.sqrt(((other.x - self.x)**2+(other.y - self.y)**2))
p1=Point(1,2)
p2=Point(4,5)
d=p1.distance(p2)
print("The distance between 2 points is :",d)
```

The distance between 2 points is : 4.242640687119285

In [ ]: ►