**Example-01:**

class A:

def \_\_init\_\_(self,x):

self.x = x

def \_\_add\_\_(self,R):

r = self.x + R.y

return r

class B:

def \_\_init\_\_(self,y):

self.y = y

c1 = A(3)

c2 = B(4)

print("Sum = ", c1+c2)

**Example-02:**

class Complex:

def \_\_init\_\_(self, r, i):

self.real = r

self.img = i

def \_\_add\_\_(self, sec):

r = self.real + sec.real

i = self.img + sec.img

return (r,i)

def \_\_str\_\_(self):

return str(self.real)+' + '+str(self.img)+'i'

c1 = Complex(5,9)

c2 = Complex(2,4)

print("Sum = ",c1+c2)

**Example-03:**

class Point:

def \_\_init\_\_(self, x=0, y=0):

self.x = x

self.y = y

def \_\_str\_\_(self):

return "({0},{1})".format(self.x, self.y)

def \_\_sub\_\_(self, other):

x = self.x - other.x

y = self.y - other.y

return Point(x, y)

p1 = Point(5, 4)

p2 = Point(4, 2)

print("Sub =",p1-p2)

**Case Study-01:**

from abc import ABC, abstractmethod

class Fruit(ABC):

def name(self):

pass

def type(self):

pass

def color(self):

pass

def taste(Self):

pass

class Apple(Fruit):

def name(self):

print("Name: Apple")

def type(self):

print("Type: Fruit")

def color(self):

print("Color: Red")

def taste(self):

print("Taste: Sour")

class Mango(Fruit):

def name(self):

print("Name: Mango")

def type(self):

print("Type: Fruit")

def color(self):

print("Color: Yellow")

def taste(self):

print("Taste: Yummy")

f = Apple()

f.name()

f.type()

f.color()

f.taste()

f1 = Mango()

f1.name()

f1.type()

f1.color()

f1.taste()

**Case Study-02:**

from abc import ABC, abstractmethod

class Vehicle(ABC):

def name(self):

pass

def model(self):

pass

def speed(self):

pass

class Car(Vehicle):

def name(self):

print("Vehicle: Car")

def model(self):

print("Model: 2019")

def speed(self):

print("Speed: Maximum 250km/h")

class MotorBike:

def name(self):

print("Vehicle: Bike")

def model(self):

print("Model: 2020")

def speed(self):

print("Speed: Maximum 120km/h")

v = Car()

v.name()

v.model()

v.speed()

v1 = MotorBike()

v1.name()

v1.model()

v1.speed()

**Task-01:**

class Distance:

def \_\_init\_\_(self,f,i):

self.foot = f

self.inches = i

def \_\_add\_\_(self,other):

f = self.foot + other.foot

i = self.inches + other.inches

return Distance(f,i)

def \_\_str\_\_(self):

return "({},{})".format(self.foot,self.inches)

d1= Distance(12,33)

d2= Distance(44,65)

print("Distance =", d1 + d2)

**Task-02:**

class Time:

def \_\_init\_\_(self,h,m,s):

self.hour = h

self.minutes = m

self.second = s

def \_\_sub\_\_(self,other):

h = self.hour - other.hour

m = self.minutes - other.minutes

s = self.second - other.second

return Time(h,m,s)

def \_\_str\_\_(self):

return "({}:{}:{})".format(self.hour,self.minutes,self.second)

time1=Time(5,56,8)

time2=Time(2,15,4)

print("Time is:",time1-time2)