MULTILEVEL INHERITANCE:

class SuperClass:

def super\_method(self):

Print(“2213711058039”)

print("Super Class method called super class")

class DerivedClass1(SuperClass):

def derived1\_method(self):

print("Derived class 1 method called derived method")

class DerivedClass2(DerivedClass1):

def derived2\_method(self):

print("Derived class 2 method called derived method2")

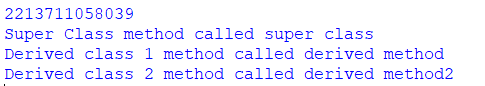
d2 = DerivedClass2()

d2.super\_method()

d2.derived1\_method()

d2.derived2\_method()

OUTPUT:



FIBONACCI SEQUENCES

def fibonacci(n):

fib\_sequence = [0, 1]

while len(fib\_sequence) < n:

fib\_sequence.append(fib\_sequence[-1] + fib\_sequence[-2])

return fib\_sequence

n = 10

result = fibonacci(n)

print(f"Fibonacci sequence of the first {n} numbers: {result}")

OUTPUT:



INHERITANCE

SINGLE INHERITANCE:

class Operation:

A=50

B=10

def add(self):

sum=self.A+self.B

print("sum of a and b is: " ,sum)

class Myclass(Operation):

C=50

D=10

def sub(self):

subt=self.C-self.D

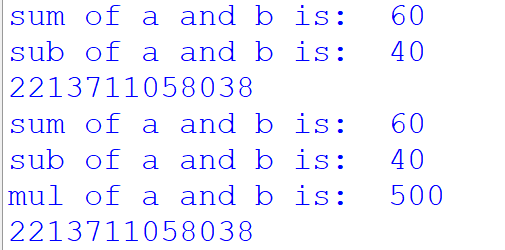
print("sub of a and b is: " ,subt)

Ob=Myclass()

Ob.add()

Ob.sub()

OUTPUT:



MULTIPLE INHERITANCE:

class Addition:

A=50

B=10

def add(self):

sum=self.A+self.B

print("sum of a and b is: " ,sum)

class Subtraction(Addition):

def sub(self):

subt=self.A-self.B

print("sub of a and b is: " ,subt)

class Multiplication(Subtraction):

def mul(self):

mul=self.B\*self.A

print("mul of a and b is: " ,mul)

Ob=Multiplication()

Ob.add()

Ob.sub()

Ob.mul()

OUTPUT:

