**INHERITANCE PROGRAMS**

**1.Write a Program to Demonstrate Single-Level Inheritance.**

#SINGLE INHERITANCE

class parent:

    def m1():

        print('IAM M1 FROM PARENT CLASS')

    def m2():

        print('IAM M2 FROM PARENT CLASS')

class child(parent):

    def m3():

        print('IAM M3 FROM CHILD CLASS')

    def m4():

        print('IAM M4 FROM CHILD CLASS')

child.m1()

child.m2()

child.m3()

child.m4()

**2.Write a Program to Demonstrate Single-Level Inheritance.**

#SINGLE INHERITANCE

class first():

    a=11

    b=11

    def add(self):

        result=self.a+self.b

        print('THE RESULT IS',result)

class second(first):

    c=66

    d=44

    def sub(self):

        result=self.c-self.d

        print('THE RESULT IS ',result)

s=second()

s.add()

s.sub()

**3.Write a Program to Demonstrate Single-Level Inheritance.**

#SINGLE INHERITANCE

class Employee:

    x = 22

    def msg(self):

        print('WELCOME TO EMPLOYEE CLASS')

class Department(Employee):

    a = 444

    b = Employee.x + 22

    def msg2(self):

        print('WELCOME TO DEPARTMENT CLASS')

    def changed(self):

        print('OLD  VALUE = ',Employee.x)

        Employee.x= Employee.x + 200

        print('NEW VALUE = ', Employee.x)

dept = Department()

dept.msg()

dept.msg2()

dept.changed()

print('\n')

print(dept.a)

print(dept.b)

**4.SINGLE INHERITANCE**

class parent:

def \_init\_(self):

self.surname='nandamuri'

self.\_\_name ='ntr'

self.\_\_age=80

self.color='red'

class child(parent):

def \_init\_(self):

super().\_init\_()

self.name='balaiah'

self.age=50

def display(self):

print(self.surname)

print(self.name)

print(self.age)

print(self.color)

c1=child()

c1.display()

**5. Write a Program to Demonstrate Multi-Level Inheritance.**

#MULTI-LEVEL INHERITANCE

class A:

    def add(self,x,y):

        z=x+y

        print('THE RESULT OF ADDITION IS ',z)

class B:

    def sub(self,x,y):

        z=x-y

        print('THE RESULT OF SUBTRACTION IS ',z)

class C:

    def mul(self,x,y):

        z=x\*y

        print('THE RESULT OF MULTIPLICATION IS ',z)

class D:

    def div(self,x,y):

        z=x/y

        print('THE RESULT OF DIVISION IS ',z)

class E(A,B,C,D):

    def mod(self,x,y):

        z=x%y

        print('THE RESULT OF MODULO-DIVISION IS ',z)

    def msg(self):

        print('-------THE END-------')

E=E()

E.add(11,22)

E.sub(33,44)

E.mul(55,66)

E.div(77,88)

E.mod(99,11)

E.msg()

6.Write a program to demonstrate hierarichal inheritance:

class person:

def getdata(self):

self.name=input('enter Name:')

self.age=int(input('enter Age:'))

self.gender=input('enter Gender:')

def display(self):

print("Name: ",self.name)

print("Age: ",self.age)

print("Gender: ",self.gender)

class student(person):

def getdata(self):

super().getdata()

self.institute=input('Name of College: ')

self.level=input('Level: ')

def display(self):

super().display()

print('Name of College: ',self.institute)

print('Level: ',self.level)

class employee(person):

def getdata(self):

super().getdata()

self.company=input('Name of Company: ')

self.sal=input('Salary: Rs. ')

def display(self):

super().display()

print('Name of Company: ',self.company)

print('Salary: Rs. ',self.sal)

s=student()

e=employee()

print('Student')

print('Enter data')

s.getdata()

print('Displaying data')

s.display();

print('Employee')

print('Enter data')

e.getdata()

print('Displaying data')

e.display()

7. **Write a Program to Demonstrate Hybrid Inheritance.**

#HYBRID INHERITANCE22

class arithmetic:

    def getdata(self):

        self.n1=int(input("\n ENTER THE FIRST NUMBER : "))

        self.n2=int(input("\n ENTER THE SECOND NUMBER : "))

class plus(arithmetic):

    def add(self):

        self.sum=self.n1 + self.n2

class minus(arithmetic):

    def sub(self):

        self.diff=self.n1 - self.n2

class result(plus,minus):

    def display(self):

        print('SUM OF {} AND {} = {} ' .format\

              (self.n1 ,self.n2, self.sum))

        print('DIFFERENCE OF {} AND {} = {}'.format

              (self.n1, self.n2, self.diff))

r1=result()

r1.getdata()

r1.add()

r1.sub()

r1.display()