1. Design a class that store the information of three students and display the same.

class students:

count = 0

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

self.name = name

self.marks = []

students.count = students.count + 1

def enterMarks(self):

for i in range(3):

m = int(input("Enter the marks of %s in %d subject: "%(self.name, i+1)))

self.marks.append(m)

def display(self):

print (self.name, "got ", self.marks)

name = input("Enter the name of Student:")

s1 = students(name)

s1.enterMarks()

s1.display()

print ("")

name = input("Enter the name of Student:")

s2 = students(name)

s2.enterMarks()

s2.display()

s2.displayCount()

2. Write a Python program to Implement the concept of inheritance using python.

#Single Inheritance

#create one parent class called ClassOne and one child class called ClassTwo to implement single inheritance.

# Base class

class Vehicle:

def Vehicle\_info(self):

print('Inside Vehicle class')

# Child class

class Car(Vehicle):

def car\_info(self):

print('Inside Car class')

# Create object of Car

car = Car()

# access Vehicle's info using car object

car.Vehicle\_info()

car.car\_info()

#Multiple Inheritance

# Parent class 1

class Person:

def person\_info(self, name, age):

print('Inside Person class')

print('Name:', name, 'Age:', age)

# Parent class 2

class Company:

def company\_info(self, company\_name, location):

print('Inside Company class')

print('Name:', company\_name, 'location:', location)

# Child class

class Employee(Person, Company):

def Employee\_info(self, salary, skill):

print('Inside Employee class')

print('Salary:', salary, 'Skill:', skill)

# Create object of Employee

emp = Employee()

# access data

emp.person\_info('Jessa', 28)

emp.company\_info('Google', 'Atlanta')

emp.Employee\_info(12000, 'Machine Learning')

#Multilevel Inheritance

# Base class

class Vehicle:

def Vehicle\_info(self):

print('Inside Vehicle class')

# Child class

class Car(Vehicle):

def car\_info(self):

print('Inside Car class')

# Child class

class SportsCar(Car):

def sports\_car\_info(self):

print('Inside SportsCar class')

# Create object of SportsCar

s\_car = SportsCar()

# access Vehicle's and Car info using SportsCar object

s\_car.Vehicle\_info()

s\_car.car\_info()

s\_car.sports\_car\_info()

3. Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).

Write a method called add which returns the sum of the attributes x and y.

Write a class method called multiply, which takes a single number parameter an and returns the product of a and MULTIPLIER.

Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.

class Numbers:

MULTIPLIER=3.5

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

self.x=x

self.y=y

def add(self):

return self.x+self.y

@classmethod

def multiply(cls,a):

return cls.MULTIPLIER\*a

@staticmethod

def subtract(b,c):

return b-c

@property

def value(self):

return (self.x,self.y)

#setter

def set\_value(self,x,y):

self.x=x

self.y=y

#deleter

def del\_value(self):

del self.x

del self.y

obj1=Numbers(10,20)

print("add",obj1.add())

# invoke class method

print("mulyiply",Numbers.multiply(10))

# invoke static method

print("subtract ",Numbers.subtract(10,5))

#invoked property

print("property ",obj1.value)

#invoked setter

print(obj1.set\_value(100,200))

#invoked property

print("property ",obj1.value)

#invoked deletor

print(obj1.del\_value())