Page No.
Date / /

```
Enter element in {}: {'N01' : 20, 'N02' : 30,
                      'N03' : 50}

Sum of values in directory : 100
```

Program: 73 : Write a program of creating class and Object :

```
Class Student:   # class name start with capital
                                            letter
   def __init__(self):

        self.name = "Hiren"
        self.age = 20
        self.marks = 90

   def talk(self):
        print("Hi i am", self.name)
        print("My age is", self.age)
        print("My marks is", self.marks)

S1 = Student()
S1.talk()
```

Output:

```
Hi i am Hiren
My age is 20
My marks is 90
```

Page No.
Date / /

Program : 74 : write a program of Constructor:

```
Class Student :
        # This is constructor

    def __init__(self, n="xyz", m=0):
        self.name = n
        self.mark = m
      # This is an instance method
    def display (self):
        print ("Hi ", self.name)
        print ("your mark is ", self.mark)

s = Student ()
s. display ()
print ("======")
s1 = Student ("KSC", 90)
s1. display ()
print ("======")
```

Output:

```
Hi xyz
your mark is 0
======

Hi KSC
your mark is 90
```

Page No.
Date / /

Program : 75 : write a program to create teacher class and store it into teacher.py

```
Class Teacher :

    def setid(self, id):
        self.id = id
    def getid(self):
        return self.id

    def setname (self, name):
        self.name = name
    def getname (self) :
        return self.name

    def setaddress (self, address):
        self.address = address
    def getaddress (self):
        return self.address

    def setsalary (self, salary):
        self. salary = salary
    def getsalary (self):
        return self.salary
```

#python program to used Teacher class:
inh.py

#using Teacher class

Page No.
Date / /

```
From teacher import Teacher
#create instance

t= Teacher()
#store data into instance

t. setid (101)
t. setname ("Amit")
t. setaddress(" Liliya road ")
t. setsalary (35000.00)

#Retrive data From instance and display

print ("id =", t.getid ())
print ("name =", t.getname())
print(" address =", t.getaddress())
 print(" salary =", t.getsalary())
```

<u>output:</u>

```
id= 101

name= Amit

address= Liliya road

salary= 35000.0
```

Page No.
Date / /

→ To create Student class by diriving it From the Teacher class :

```
From teacher import Teacher

Class Student(Teacher):

        def setmarks(self, marks):
            self.marks = marks

        def getmarks(self):
            return self.marks

s = Student()
s.setid(100)
s.setname("Rakesh")
s.setaddress("Chakargadh road")
s.setmarks(89)

print("Id=", s.getid())
print("Name=", s.getname())
print("Address=", s.getaddress())
print("Marks =", s.getmarks())
```

Output:

```
Id = 100
Name = Rakesh
Address = Chakargadh road
Marks = 89
```

Page No.
Date / /

program: 76: write a program of single inheritance:

```
#single inheritance

class Bank(Object):
        cash = 10000000
        @classmethod
        def available_cash(cls):
                print(cls.cash)

class AndhraBank(Bank):
        pass

class StateBank(Bank):
        cash = 20000000
        @classmethod
        def available_cash(cls):
                print(cls.cash + Bank.cash)

a = AndhraBank()
a.available_cash()

s = StateBank()
s.available_cash()
```

Output:

```
10000000
30000000
```

Page No.
Date / /

program : 77 : write a program of Multiple Inheritance;

```
#multiple Inheritance

class Father:
        def height(self):
                print("Height is 6.0 Foot")

Class Mother:
        def color(self):
                print("color is Brown")

Class Child(Father, Mother):
        def age(self):
                print("Age is 19")

C=Child()
print("Child's inherited qualities")
C.height()
C.color()
C.age()
```

Output:

```
Child's inherited qualities

Height is 6.0 Foot
color is Brown
Age is 19
```

Page No.
Date / /

program: 78: write a program OF polymorphis

```
#overloading the + operator
# using + on integer to add sum

print(10+20)

#using + on string

s1="ksc"
s2="pac"

print(s1+s2)

#using + on list

a=[10,20,30,40]
b=[5,15,25]

print(a+b)
```

Output:

```
30

KSCPAC

[10,20,30,40, 5, 15, 25]
```

Page No.
Date / /

program : 77 : write a program OF private method :

```
class Car:

    def __init__(self):

        self.__updateSoftware()

    def drive(self):
        print("Driving")

    def __updateSoftware(self):
        print("updating software")

redcar = Car()
redcar.drive()

#redcar.__updateSoftware  it will give
                error because method not call
                                directly
```

Output :

```
updating software

Driving
```