In [29]:

```
#1.Python program to find volume and surface area of Cylinder (V=pi*r*r*h, SA=2*pi*r*h)
#using class and objects. Create a constructor to initialize the objects and print the
#with 2 decimal points precision.( Finding Vol and SA using class and object)

import math
class cylinder:
    def __init__(self,radius):
        self.radius=radius
    def volume(self):
        return math.pi*(self.radius**2)*h
    def surface(self):
        return (2*math.pi*(self.radius*h)+(math.pi*self.radius**2))
r=float(int(input("Enter radius : ")))
h=float(int(input("Enter height : ")))
obj=cylinder(r)
obj=cylinder(h)
print("volume of cylinder:",(obj.volume()))
print("surface area of cylinder:",(obj.surface()))
```

```
Enter radius : 4
Enter height : 6
volume of cylinder: 678.5840131753953
surface area of cylinder: 339.29200658769764
```

In [28]:

```
2.#Create a class Student with constructor, setdata() and dispdata() for encapsulating
#rollno, name, mark1, mark2 into it. Create three objects obj1, obj2, obj3 from class
#Student. Set rollno, name, mark1, mark2 for 3 students using setdata() and display
#them using dispdata().
#(simple class and object with constructor)

class student:

    studCount = 0
    def __init__(self,name, rollno, mark1,mark2):
        self.name = name
        self.rollno = rollno
        self.mark1 = mark1
        self.mark2 = mark2
        student.studCount += 1

    def setdata(self):
        print("Total students %d" % student.studCount)
    def displaydata(self):
        print ("Name:",self.name, "rollno:",self.rollno, "mark1:",self.mark1, "mark2:",

stud1 = student("Zara", 2,10,11)

stud2 = student("Manni", 5,11,12)

stud3 = student("nani",6,16,13)
stud1.displaydata()
stud2.displaydata()
stud3.displaydata()
print("Total student %d" % student.studCount)
```

```
Name :  Zara rollno:  2 mark1: 10 mark2: 11
Name :  Manni rollno:  5 mark1: 11 mark2: 12
Name :  nani rollno:  6 mark1: 16 mark2: 13
Total student 3
```

In [3]:

```
3.#Create a parent class Person with constructor(name, idnumber), display() to display
#name and idnumber and child class Employee with constructor(name, idnumber
#salary, post) and display() to display name, idnumber, salary and post. Create object
#from parent to pass name and idnumber as parameter and display them. Create
#objects a and b of Employee to pass name, idnumber, salary and post and display
#them (single inheritance)


class Person:
  def __init__(self, name, idnumber):
    self.name=name
    self.idnumber=idnumber
  def display(self):
    print(f'Name: {self.name}\nIDNumber: {self.idnumber}')

class Employee(Person):
  def __init__(self,name, idnumber, salary, post):
    self.salary=salary
    self.post=post
    super(Employee,self).__init__(name, idnumber)
  def display(self):
    super(Employee,self).display()
    print(f'Salary: {self.salary}\nPost: {self.post}')

x=Person('hi',346)
print('Parent Class')
x.display()
a=Employee('vinod',984,40000,'Teacher')
print('\nChild Class')
a.display()
b=Employee('how have you been?',254,50000,'Mannager')
print('\nChild Class')
b.display()
```

```
Parent Class
Name: hi
IDNumber: 346

Child Class
Name: vinod
IDNumber: 984
Salary: 40000
Post: Teacher

Child Class
Name: how have you been?
IDNumber: 254
Salary: 50000
Post: Mannager
```

In [19]:

```
#4.Create a parent class student in which a method getStudent() is defined to get rollr
#and name of the student. Create a child class called test in which a method
#getMarks() is defined to get maths and science marks. Create a grandchild class
#called marks in which display() is defined to display all the details
#rollno,name,maths marks, science marks and average marks (of science and maths)
#(Multilevel inheritance problem)

class Student:
  def getStudent(self,rollno,name):
    self.n = name
    self.r = rollno

class Test(Student):
  def getMarks(self,maths,science):
    self.m1 = maths
    self.s = science

class Marks(Test):
  def display(self):
    print("Name : {0}\n RollNo : {1}\nMaths marks : {2}\nScience Marks : {3}\nAverage :
m = Marks()
m.getStudent(input("Enter the rollno. : "), input("Enter the name : "))
m.getMarks(int(input("Enter the Maths marks : ")),int(input("Enter the Science marks :
m.display()
```

```
Enter the rollno. : 1
Enter the name : vinod
Enter the Maths marks : 100
Enter the Science marks : 200
Name : vinod
 RollNo : 1
Maths marks : 100
Science Marks : 200
Average : 150.0
```

In [20]:

```
#Create classes India and USA . Each class has captial() , language() and currency()
#methods. They print information of capital, language and currency of the respective
#country (capital() in India class should print 'New Delhi' capital() in USA should
#print 'Washington DC' similarly for language() of India Hindi and English ,
#Language() of USA should print English. Similarly currency() should print Rupee
#for India and Dollar for USA using Polymorphism by creating two objects obj1 and
#obj2 for India and USA respectively. Print the all the information using for
#loop(Polymorphism problem)

class India():
  def __init__(self,capital,language,currency):
    self.capital=capital
    self.language=language
    self.currency=currency
  def capitale(self):
    print("INDIA's\nCapital = ",self.capital)
  def lang(self):
    print("Language = ",self.language)
  def curr(self):
    print("Currency = ",self.currency)



class USA():
  def __init__(self,capital,language,currency):
    self.capital=capital
    self.language=language
    self.currency=currency
  def capitale(self):
    print("USA's\nCapital = ",self.capital)
  def lang(self):
    print("Language = ",self.language)
  def curr(self):
    print("Currency = ",self.currency)

obj1=India("Delhi","Hindi and English","Rupee")
obj2=USA("Washington DC","Hindi and English","Dollar")
obj=[obj1,obj2]
for i in obj:
  i.capitale()
  i.curr()
  i.lang()
  print("\n")
```

```
INDIA's
Capital =  New Delhi
Currency =  Rupee
Language =  Hindi and English


USA's
Capital =  Washington DC
Currency =  Dollar
Language =  Hindi and English
```

In [ ]:

```
1
```