In [2]:
```
#Q1 Create a class Product with Product ID, Name, Quantity, and Rate as ir
#variables from user; calculate the Price of each product and Total Amount
#per product and Total Amount back to user.
class Product:
    def __init__(self,pid,name,qty,rate):
        self.product_id=pid
        self.product_name=name
        self.product_quantity=qty
        self.product_rate=rate

    def Display(self):
        print("Product Id",pid)
        print("Product Name",name)
        print("Product Quantity",qty)
        print("Rate",rate)
        total=quantity*rate
        print("Total Price",total)
```

In [3]:
```
pid=int(input("Enter Product Id: "))
name=input("Enter Product Name: ")
quantity=int(input("Enter Product Quantity: "))
rate=int(input("Enter Product Rate: "))
P1=Product(pid,name,quantity,rate)
P1.Display()
```

```
Enter Product Id: 101
Enter Product Name: book
Enter Product Quantity: 10
Enter Product Rate: 4
Product Id 101
Product Name book
Product Quantity 10
Rate 4
Total Price 40
```

In [4]:

```
#Q2 Create a class Time with instance variable Hours, Minutes, and Seconds
#user and initialize using constructor. Overload the '+'operator to add tw
#in Python.
class Time:
    def __init__(self,h,m,s):
        self.hrs = h
        self.mins = m
        self.sec = s

    def __add__(self,Ob2):
        Ob3 = Time(0,0,0)
        Ob3.hrs = self.hrs + Ob2.hrs
        Ob3.mins = self.mins + Ob2.mins
        Ob3.sec = self.sec + Ob2.sec

        if Ob3.sec >=60:
            Ob3.sec = Ob3.sec - 60
            Ob3.mins = Ob3.mins + 1

        if Ob3.mins >=60:
            Ob3.mins = Ob3.mins - 60
            Ob3.hrs = Ob3.hrs + 1
        return Ob3

    def Display(self):
        print("Hours: ",self.hrs)
        print("Minutes: ",self.mins)
        print("Seconds: ",self.sec)

T1 = Time(3,40,45)
T2 = Time(4,30,20)
T3 = Time(0,0,0)

T3 = T1+T2
T3.Display()

```

```
Hours:  8
Minutes:  11
Seconds:  5
```

In [4]:

```
#Q3 Create a class Student with data members roll no &amp;amp; name, add appr
#student details. Inherit 3 classes from Student as Science, Arts &amp;amp; C
#subjects as data members. Arts class contains marks of two subjects as d
#two subjects as data members. Create Python program to achieve Hierarchi
class Student:
    def __init__(self,rollno,name):
        self.rno=rollno
        self.sname=name

class Science(Student):
    def __init__(self,rollno,name,phy,che):
        Student.__init__(self,rollno,name)
        self.physics=phy
        self.chemistry=che

    def Sdisplay(self):
        print("Roll Number of Student: ",self.rno)
        print("Name of Student: ",self.sname)
        print("Physics Marks: ",self.physics)
        print("Chemistry Marks: ",self.chemistry)

class Arts(Student):
    def __init__(self,rollno,name,eng,eco):
        Student.__init__(self,rollno,name)
        self.english=eng
        self.economics=eco

    def Adisplay(self):
        print("Roll Number of Student: ",self.rno)
        print("Name of Student: ",self.sname)
        print("English Marks: ",self.english)
        print("Economics Marks: ",self.economics)

class Commerce(Student):
    def __init__(self,rollno,name,acc,bank):
        Student.__init__(self,rollno,name)
        self.accounts=acc
        self.banking=bank

    def Cdisplay(self):
        print("Roll Number of Student: ",self.rno)
        print("Name of Student: ",self.sname)
        print("Accounts Marks: ",self.accounts)
        print("Banking Marks: ",self.banking)
```

In [5]:
```
S1=Science (1,"Amar", 89,78)
S1.Sdisplay()

A1=Arts (2, "Akbar", 67, 80)
A1.Adisplay()

C1=Commerce (3, "Ankush", 92, 90)
C1.Cdisplay()
```

```
Roll Number of Student:  1
Name of Student:  Amar
Physics Marks:  89
Chemistry Marks:  78
Roll Number of Student:  2
Name of Student:  Akbar
English Marks:  67
Economics Marks:  80
Roll Number of Student:  3
Name of Student:  Ankush
Accounts Marks:  92
Banking Marks:  90
```

In [6]:
```
#Q4 Create an abstract base class called Shape with two instance variables
#used to compute the area. Derive two specific classes called Triangle and
#class constructor through derived class constructor for initializing data
#compute and display area of shapes. Create python program to implement ru
from math import pi
class Shape:
    def __init__(self,name):
        self.name=name

    def area(self):
        pass

class Square(Shape):
    def __init__(self, length):
        super().__init__("Square")
        self.length = length

    def area(self):
        return self.length**2

class Circle(Shape):
    def __init__(self, radius):
        super().__init__("Circle")
        self.radius=radius

    def area(self):
        return pi*self.radius**2
```

In [7]:

```
a = Square(4)
b = Circle(7)

print (a.area())
print(b.area())
```

```
16
153.93804002589985
```