In [1]:
```
#12.1
#area of rectangle using class and object
class Rectangle:
    def area(self,l,b):
        return l*b
    def pmtr(self,l,b):
        return (l+b)*2

r1=Rectangle()
r2=Rectangle()

l1=int(input("ENTER LENGTH OF FIRST RECTANGLE:" ))
b1=int(input("ENTER BREADTH OF FIRST RECTANGLE: "))

l2=int(input("ENTER LENGTH OF SECOND RECTANGLE:" ))
b2=int(input("ENTER BREADTH OF SECOND RECTANGLE:" ))

print("PERIMETER OF FIRST: ",r1.pmtr(l1,b1))
print("PERIMETER OF SECOND: ",r1.pmtr(l2,b2))

a1=r1.area(l1,b1)
a2=r2.area(l2,b2)

if a1>a2:
    print("FIRST RECTANGLE IS LARGER IN AREA AND THE DIFFERENCE IS,",(a1-a2))
else:
    print("SECOND RECTANGLE IS LARGER IN AREA AND THE DIFFERENCE IS,",(a2-a1))
```

```
ENTER LENGTH OF FIRST RECTANGLE:10
ENTER BREADTH OF FIRST RECTANGLE: 5
ENTER LENGTH OF SECOND RECTANGLE:5
ENTER BREADTH OF SECOND RECTANGLE:5
PERIMETER OF FIRST:  30
PERIMETER OF SECOND:  20
FIRST RECTANGLE IS LARGER IN AREA AND THE DIFFERENCE IS, 25
```

In [4]:
```
#12.2
#details of student using constructor
class student:
    def __init__(self,r,n,c):
        self.rno=r
        self.name=n
        self.cls=c
    def disp(self):
        print("ROLL NO,NAME,CLASS=> ",self.rno,",",self.name,",",self.cls)
nm=input("Enter name: ")
rn=int(input("Enter roll no: "))
cl=int(input("Enter cls: "))
stud=student(rn,nm,cl)
stud.disp()
```

```
Enter name: Ghanasyam
Enter roll no: 11
Enter cls: 10
ROLL NO,NAME,CLASS=>  11 , Ghanasyam , 10
```

In [6]:
```
#12.3
#bank account,deposit,debit

class bankaccount:
    def __init__(self,accno,name,typ,bal):
        self.accnum=accno
```

```
        self.nm=name
        self.ty=typ
        self.ba=bal
    def deposit(self,damnt):
        self.ba=self.ba+damnt
        print("SUCCESSFULLY DEPOSITED...,Your current balance is ",self.ba)
    def withdraw(self,wamnt):
        self.ba=self.ba-wamnt
        print("SUCCESSFULLY DEBITED...,Your current balance is ",self.ba)

an=int(input("Enter account number: "))
na=input("Enter name: ")
t=input("Enter account type: ")
ba=int(input("Enter first deposit(min-500):"))
p1=bankaccount(an,na,t,ba)

while(1):

    ch=int(input("If you want to withdraw press 1 ,deposite press 2 ,unless any key.
    if ch==1:
        wamount=int(input("Enter amount to debit.."))
        if p1.ba<wamount+500:
            print("sorry You dont have enough balance to debit...")
        else:

            p1.withdraw(wamount)
    elif ch==2:
        damount=int(input("Enter amount to deposite.."))
        p1.deposit(damount)
    else:
        print("Thank you come again...")
        break
```

```
Enter account number: 100
Enter name: shaym
Enter account type: savings
Enter first deposit(min-500):5000
If you want to withdraw press 1 ,deposite press 2 ,unless any key...5
Thank you come again...
```

In [1]:

```
#12.4
#publisher,book,title,author

class publisher:
    def __init__(self,pname):
        self.pubname=pname
    def dis(self):
        print("Publisher name=",self.pubname)
class book(publisher):
    def __init__(self,pname,bname,author):

        self.title=bname
        self.auth=author
        publisher.__init__(self,pname)
    def display(self):
        print("Title=",self.title)
        print("Author=",self.auth)
class python(book):
    def __init__(self,pname,titl,au,price,np):

        self.rate=price
        self.nop=np
```

```
        book.__init__(self,pname,titl,au)
    def disp(self):

        print("Rate=",self.rate)
        print("Number of pages=",self.nop)

p=input("Enter publisher name:")
bk=input("Enter book name: ")
author=input("Enter author name: ")
rt=int(input("Enter rate of book:"))
nup=int(input("Enter number of pages:"))
a=python(p,bk,author,rt,nup)
a.dis()
a.display()
a.disp()
```

```
Enter publisher name:a
Enter book name: s
Enter author name: d
Enter rate of book:100
Enter number of pages:100
Publisher name= a
Title= s
Author= d
Rate= 100
Number of pages= 100
```

In [ ]:

```
#12.5
#
class person:
    def __init__(self,name,code):
        self.nm=name
        self.cod=code
class account(person):


class experience(person):
```

In [15]:

```
#12.6
#example of operator overloading ">"
class Rectangle:
    def __init__(self,l,b):
        self.a=l*b
```

```
    def __gt__(self,other):
        if self.a>other.a:
            print("First is larger")

        else:
             print("Second is larger")


l1=int(input("ENTER LENGTH OF FIRST RECTANGLE:" ))
b1=int(input("ENTER BREADTH OF FIRST RECTANGLE: "))

l2=int(input("ENTER LENGTH OF SECOND RECTANGLE:" ))
b2=int(input("ENTER BREADTH OF SECOND RECTANGLE:" ))

r1=Rectangle(l1,b1)
r2=Rectangle(l2,b2)

r1>r2
#if l1>l2:
#    print("First is larger")
#else:
#    print("Second is larger")
```

```
ENTER LENGTH OF FIRST RECTANGLE:5
ENTER BREADTH OF FIRST RECTANGLE: 5
ENTER LENGTH OF SECOND RECTANGLE:10
ENTER BREADTH OF SECOND RECTANGLE:5
Second is larger
```

In [25]:

```
#12.7
#example of operator overloading "+"
#to add 2 times

class time:
    def __init__(self,hr,mint,sec):
        self.__h=hr
        self.__m=mint
        self.__s=sec
    def __add__(self,other):
        ss=self.__s+other.__s
        if ss>=60:
            mm=1
            ss=ss-60
        mm=mm+self.__m+other.__m
        if mm>=60:
            hh=1
            mm=mm-60
        hh=hh+self.__h+other.__h
        if hh>=13:
            hh=hh-12
        print("Added Time=>",hh,"hours:",mm,"minutes:",ss,"seconds")




h1=int(input("Enter hours for first:"))
m1=int(input("Enter minutes for first"))
s1=int(input("Enter seconds for first:"))

h2=int(input("Enter hours for second:"))
m2=int(input("Enter minutes for second"))
s2=int(input("Enter seconds for second:"))
```

```
t1=time(h1,m1,s1)
t2=time(h2,m2,s2)

t1+t2
```

```
Enter hours for first:9
Enter minutes for first30
Enter seconds for first:55
Enter hours for second:3
Enter minutes for second35
Enter seconds for second:10
Added Time=> 1 hours: 6 minutes: 5 seconds
```

In [ ]: