**CO4 PROGRAMS**

**Programme1:-**

**Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.**

class rectangle:

def \_\_init\_\_(self,l,b):

self.l=l

self.b=b

def area(self):

area=self.l\*self.b

print("Area of rectangle :",area)

return(area)

def perimeter(self):

per=2\*(self.l+self.b)

print("Perimeter of rectangle :",per)

return(per)

r1=rectangle(6,7)

r2=rectangle(8,9)

a=r1.area()

r1.perimeter()

b=r2.area()

r2.perimeter()

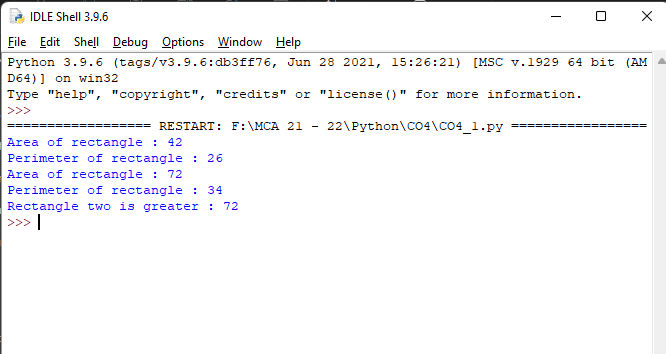
if(a>b):

print("Rectangle one isgreater :",a)

else:

print("Rectangle two is greater :",b)

**OUTPUT**



**Programme 2:-**

**Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank**

class bank:

bal=0

def \_\_init\_\_(self,accno,name,ac\_type,bal):

self.accno=accno

self.name=name

self.ac\_type=ac\_type

self.bal=bal

def display(self):

print("\nAccount info:")

print("Account number:",self.accno)

print("Account name:",self.name)

print("Account type:",self.ac\_type)

print("Account balance:",self.bal)

def deposit(self):

dep=int(input("Enter amount deposit:"))

self.bal=self.bal+dep

def withdraw(self):

w=int(input("Enter amount withdraw:"))

if w > self.bal:

print("Insufficient Balance")

else:

self.bal=self.bal-w

print("Rs",w,"Successfully Withdrawn")

acc\_no=int(input("Enter Account Number:"))

acc\_name=input("Enter name:")

acc\_type=input("Enter account type(savings/current):")

balance=int(input("Enter initial balance:"))

b1=bank(acc\_no,acc\_name,acc\_type,balance)

while(1):

print("\n1.Account info\n2.Deposit\n3.Withdraw\n4.Exit")

opt=int(input("Select your option:"))

if opt == 1:

b1.display()

elif opt == 2:

b1.deposit()

elif opt == 3:

b1.withdraw()

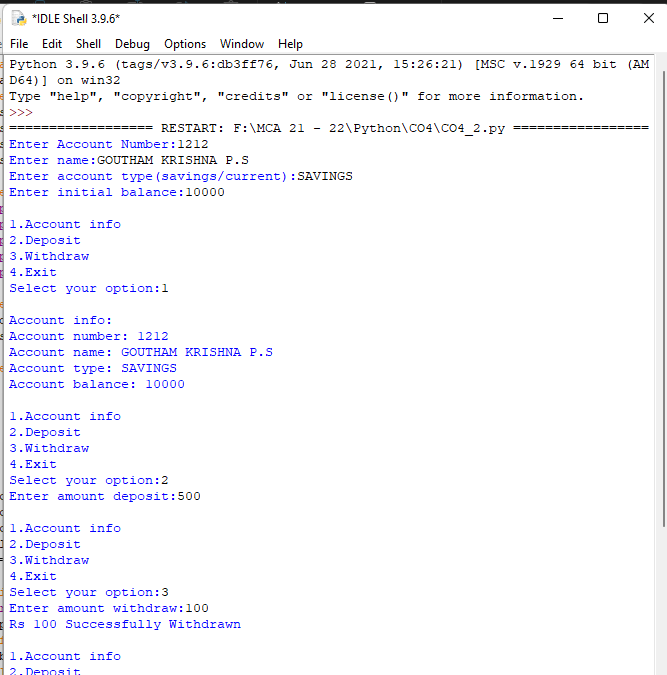
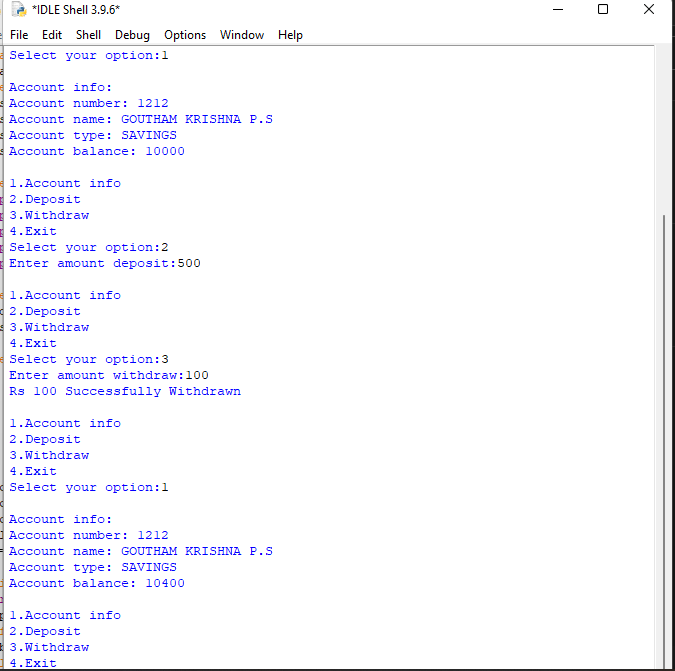
elif opt == 4:

print("Exit")

break

else:

print("Invalid Option")

**OUTPUT**

**Programme 3:-**

**Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.**

class rectangle:

def \_\_init\_\_(self,l,b):

self.\_\_length=l

self.\_\_breadth=b

def area(self):

self.area=self.\_\_length\*self.\_\_breadth

print("Area=",self.area)

def \_\_lt\_\_(self,second):

if self.area < second.area:

return True

else:

return False

print(" Rectangle 1")

len1=int(input("Enter length:"))

bread1=int(input("Enter breadth:"))

obj1=rectangle(len1,bread1)

obj1.area()

print(" Rectangle 2")

len2=int(input("Enter length:"))

bread2=int(input("Enter breadth:"))

obj2=rectangle(len2,bread2)

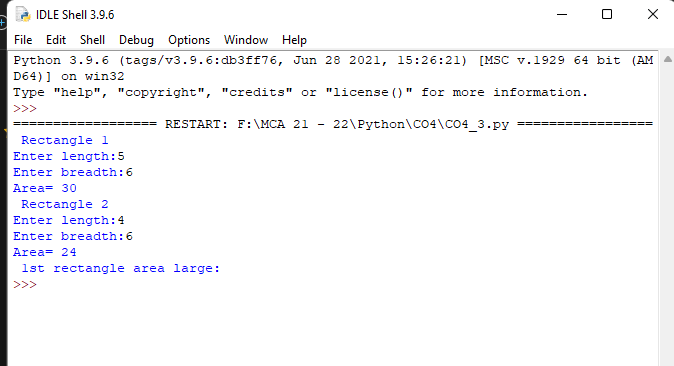
obj2.area()

if obj1 < obj2 :

print(" 2nd rectangle area large:")

else:

print(" 1st rectangle area large:")

**OUTPUT**

**Programme 4:-**

**Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.**

class time:

def \_\_init\_\_(self,hour,minute,second):

self.\_\_hour=hour

self.\_\_minute=minute

self.\_\_second=second

def \_\_add\_\_(self,t2):

h=self.\_\_hour+t2.\_\_hour

m=self.\_\_minute+t2.\_\_minute

if(m>60):

q=int(m/60)

r=m%60

h=h+q

m=r

s=self.\_\_second+t2.\_\_second

if(s>60):

q1=int(s/60)

r1=s%60

m=m+q1

s=r1

return(h,m,s)

print("Enter Time 1:")

h1=int(input("Hour:"))

m1=int(input("Minute:"))

s1=int(input("Second:"))

t1=time(h1,m1,s1)

print("Enter Time 2:")

h2=int(input("Hour:"))

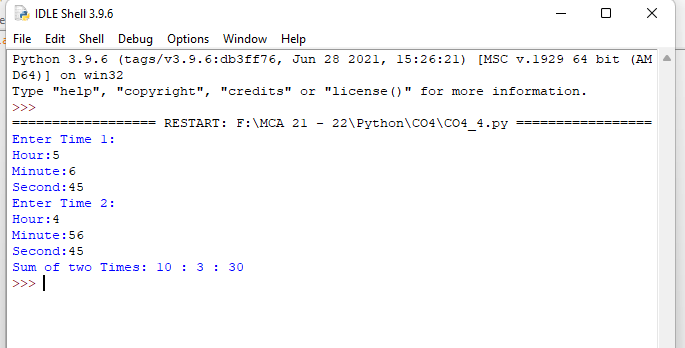
m2=int(input("Minute:"))

s2=int(input("Second:"))

t2=time(h2,m2,s2)

h,m,s=t1+t2

print("Sum of two Times:",h,":",m,":",s)

**OUTPUT**

**Programme 5:-**

**Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no\_of\_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.**

class publisher:

def \_\_init\_\_(self,title,author):

self.title=title

self.author=author

def display(self):

print("Title:",self.title)

print("Author:",self.author)

class book(publisher):

def \_\_init\_\_(self,price,no\_of\_page):

self.price=price

self.no\_of\_page=no\_of\_page

def display(self):

print("Price:",self.price)

print("No. of Pages:",self.no\_of\_page)

class python(book):

def \_\_init\_\_(self,title,author,price,no\_of\_page):

publisher.\_\_init\_\_(self,title,author)

book.\_\_init\_\_(self,price,no\_of\_page)

def display(self):

print("Title:",self.title)

print("Author:",self.author)

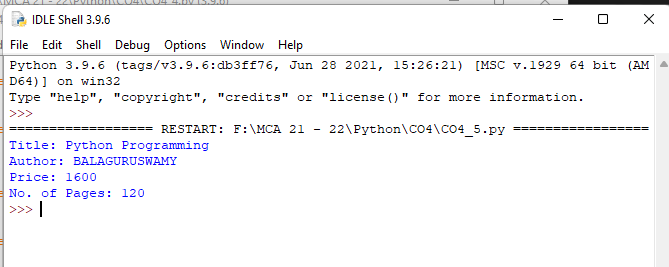
print("Price:",self.price)

print("No. of Pages:",self.no\_of\_page)

p=python("Python Programming","BALAGURUSWAMY",1600,120)

p.display()

**OUTPUT**

****