**Co4 programs**

**Program 1**

1. 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.breadth=b

self.length=l

def area(self):

return self.breadth\*self.length

def perimeter(self):

return 2\*(self.breadth+self.length)

r1=rectangle(10,20)

r2=rectangle(3,5)

print("Area of rectangle 1 :",r1.area())

print("Area of rectangle 2 :",r2.area())

print("Peimeter of rectangle 1:",r1.perimeter())

print("Peimeter of rectangle 2:",r2.perimeter())

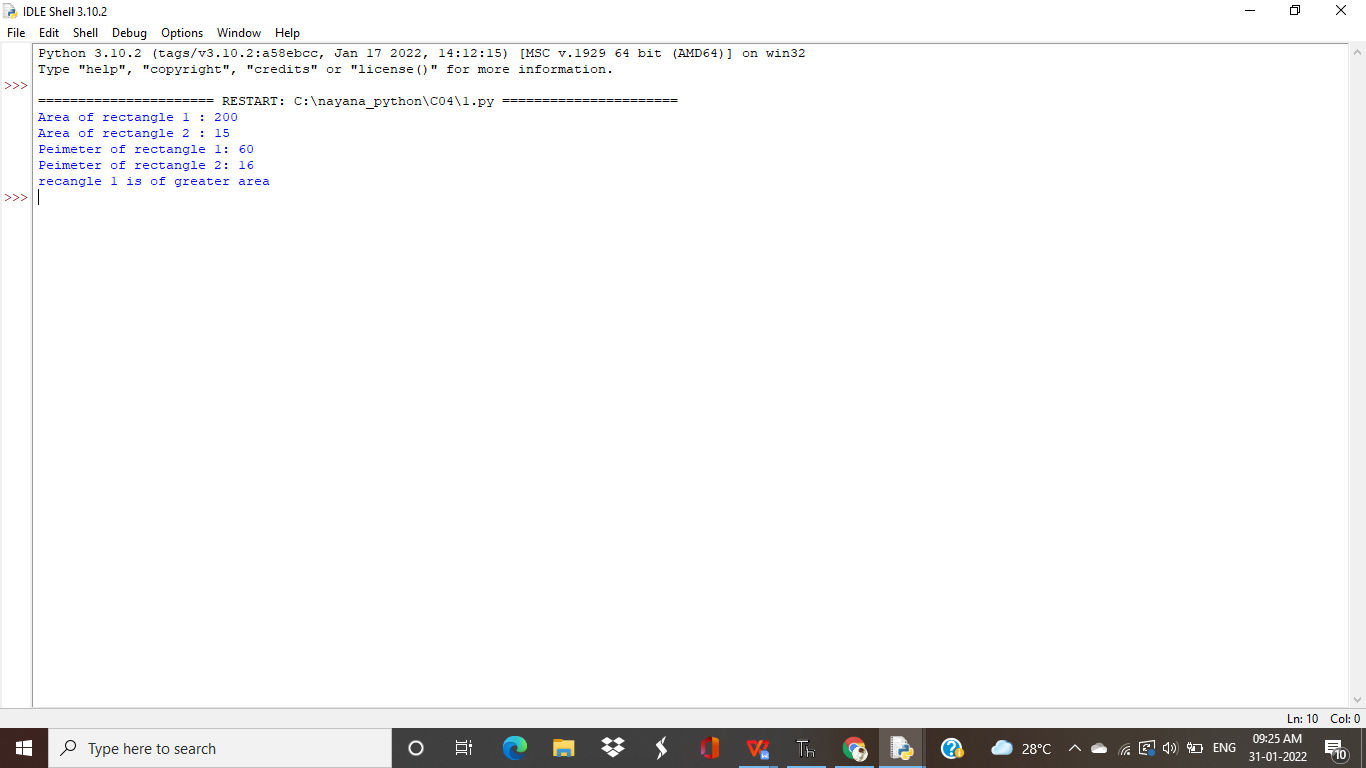
if(r1.area()>r2.area()):

print("recangle 1 is of greater area")

else:

print("recangle 2 is of greater area")

**Output**



**program 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:

balance=0

def \_\_init\_\_(self,accountno,name,accounttype,balance):

self.accountno=accountno

self.name=name

self.accounttype=accounttype

self.balance=balance

def accountinformation(self):

print("\n --ACCOUNT INFORMATION--\n")

print("Account Number:",self.accountno)

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

print("Account Type:",self.accounttype)

print("Account Balance:",self.balance,".00")

print("------------------------")

def deposit(self):

deposit=int(input("\n Enter the Amount to Deposit: "))

print("Rs.",deposit,"Deposited Successfully...")

print("------------------------")

self.balance=self.balance+deposit

def withdraw(self):

withdraw=int(input("\n Enter the Amount to Withdraw: "))

if withdraw > self.balance:

print("Your Account has Insufficient Balance...")

print("------------------------")

else:

self.balance=self.balance-withdraw

print("Rs.",withdraw,"Withdrawn Successfully...")

print("------------------------")

print(" Enter the Details of your Bank Account")

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

acc\_name=input("Enter the Name:")

acc\_type=input("Enter the Account type-(Savings/Current):")

balance=int(input("Enter the Initial Balance:"))

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

while(1):

print("\n --WELCOME TO PYTHON BANK--")

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

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

if opt == 1:

obj.accountinformation()

elif opt == 2:

obj.deposit()

elif opt == 3:

obj.withdraw()

elif opt == 4:

print("Exited")

print(" Thank You Visit Again....\n")

print("------------------------")

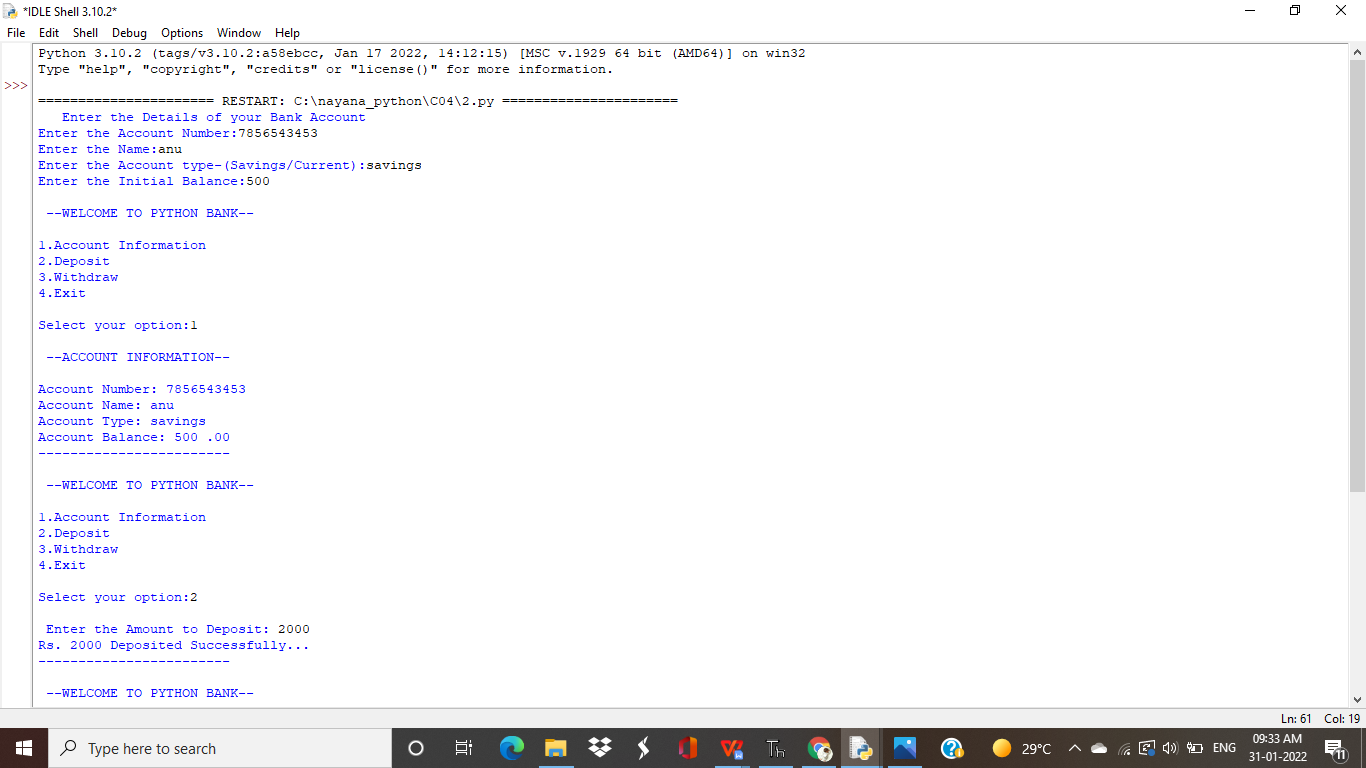
break

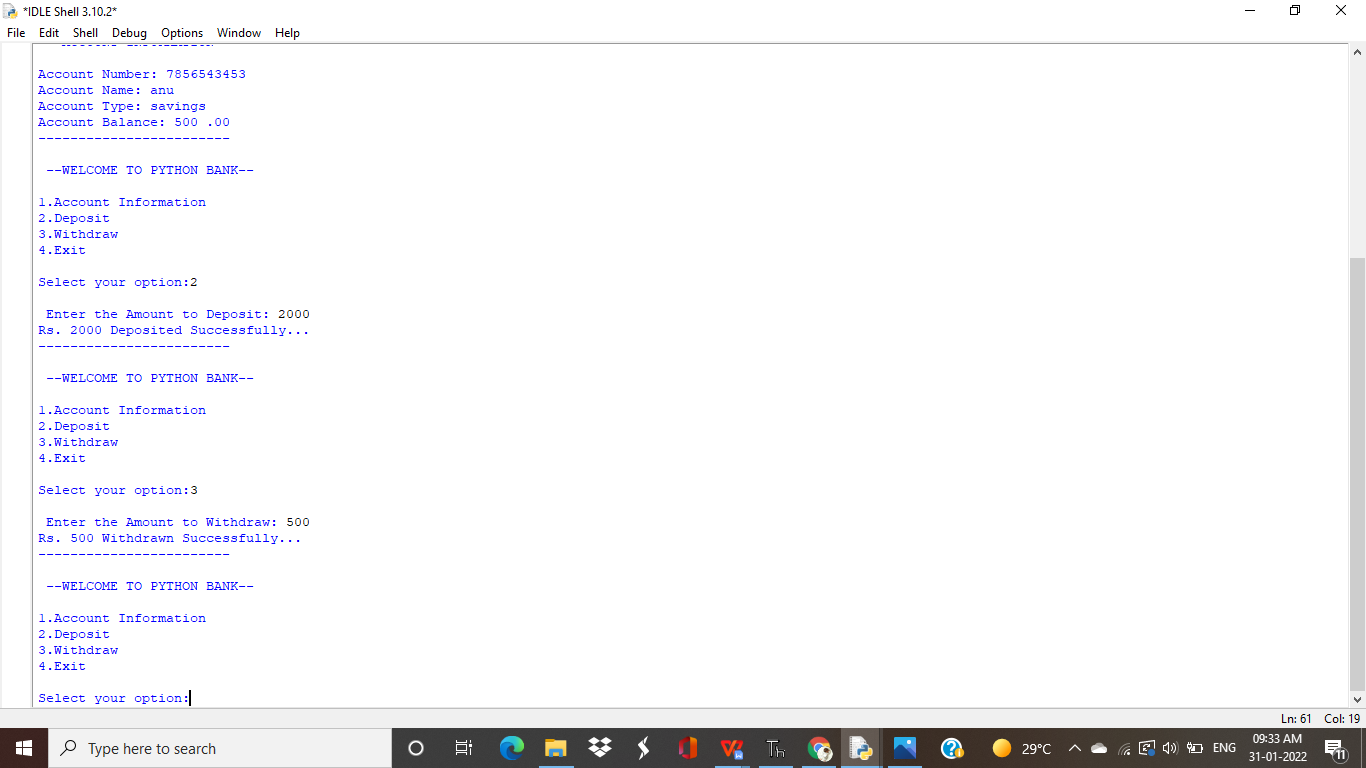
else:

print("Invalid Option")

print("------------------------")

**Output**

****

****

**Program 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 of Rectangle: ",self.area)

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

if self.area < second.area:

return True

else:

return False

print("Rectangle 1")

length1=int(input("Enter the length:"))

breadth1=int(input("Enter the breadth:"))

obj1=rectangle(length1,breadth1)

obj1.area()

print("\nRectangle 2")

length2=int(input("Enter the length:"))

breadth2=int(input("Enter the breadth:"))

obj2=rectangle(length2,breadth2)

obj2.area()

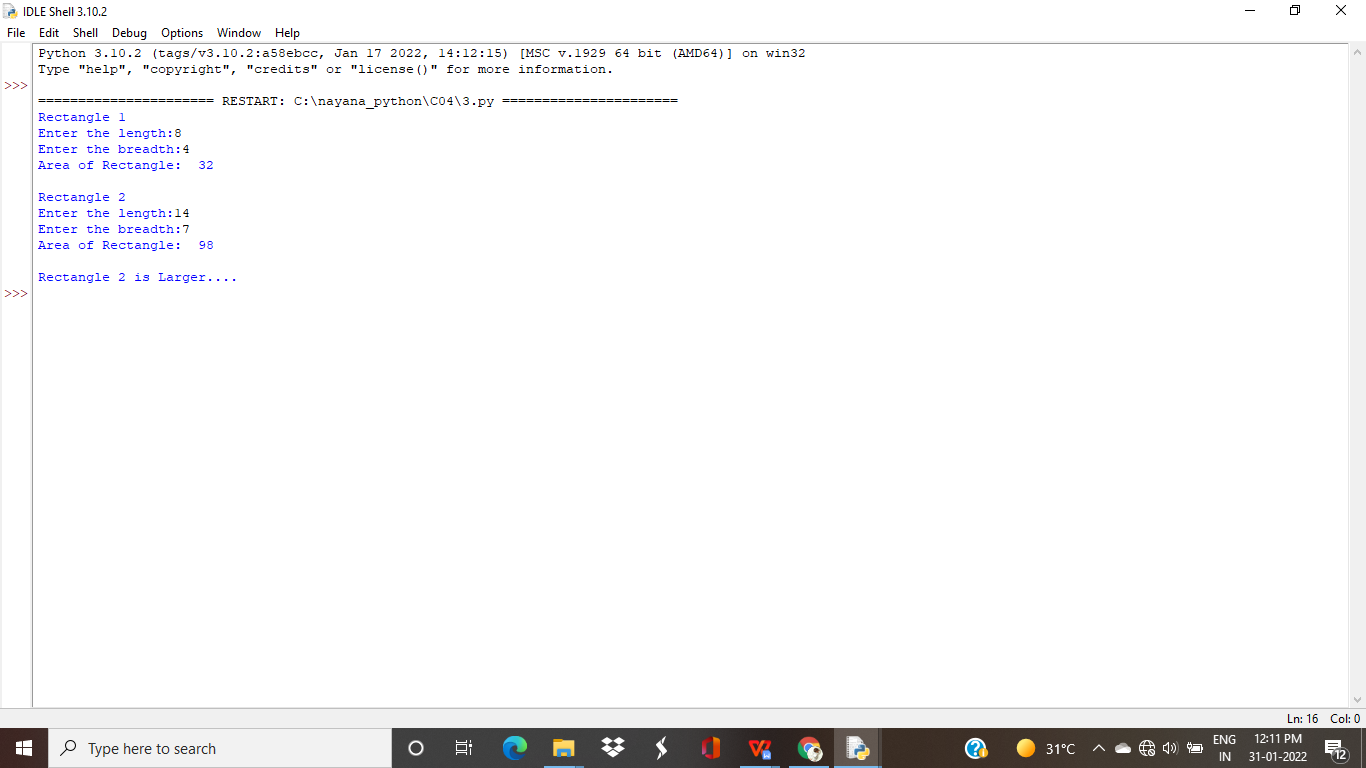
if obj1 > obj2 :

print("\nRectangle 1 is Larger.....")

else:

print("\nRectangle 2 is Larger....")

**Output**

****

**Program 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,a2):

second=self.\_\_second+a2.\_\_second

minute=self.\_\_minute+a2.\_\_minute

hour=self.\_\_hour+a2.\_\_hour

if(second>60):

second=second-60

minute=minute+1

if(minute>60):

minute=minute-60

hour=hour+1

return hour,minute,second

print("Enter time1:")

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

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

s1=int(input("second"))

t1=Time(h1,m1,s1)

print("Enter time2:")

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

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

s2=int(input("second"))

t2=Time(h2,m2,s2)

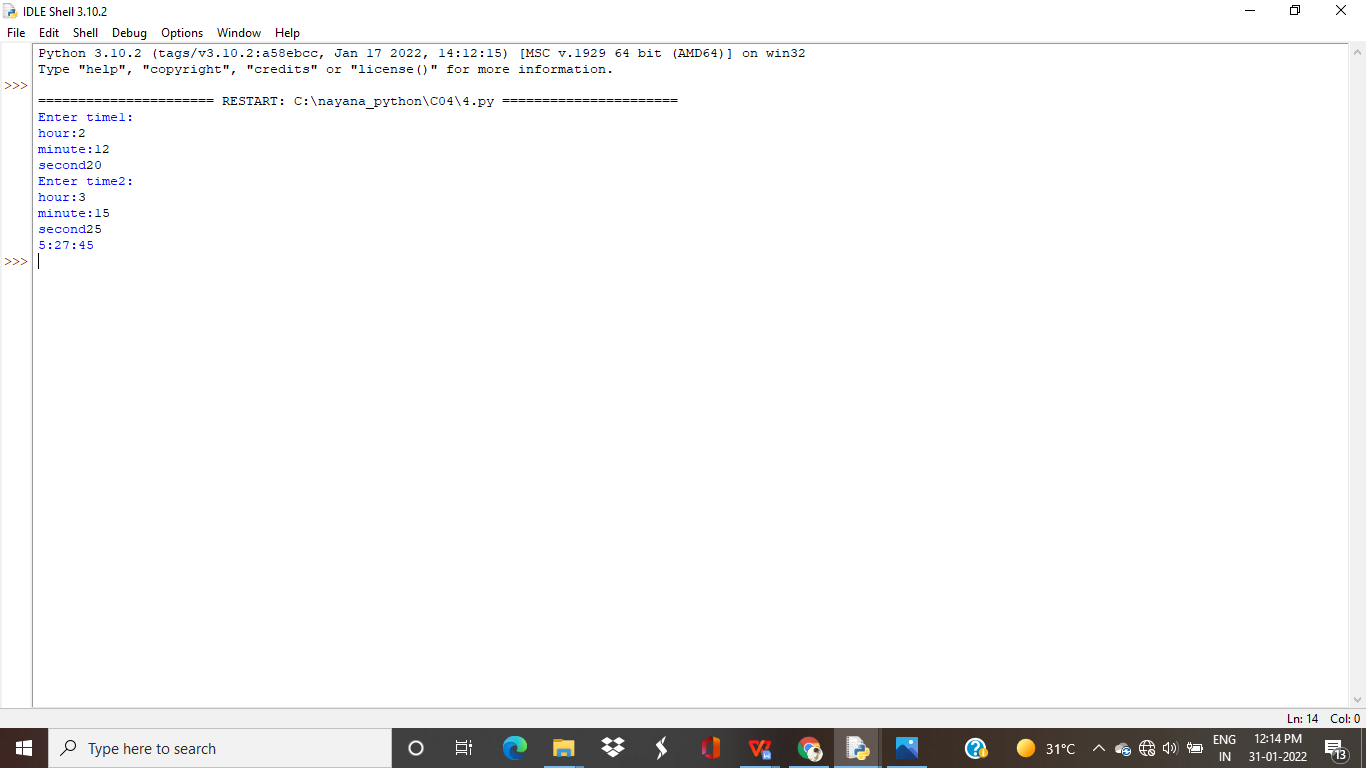
hr,min,sec=t1+t2

print(hr,end=":")

print(min,end=":")

print(sec,end=" ")

**Output**

****

**Program 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,pn):

self.publishername=pn

def display(self):

print("Publisher Name:",self.publishername)

class book(publisher):

def \_\_init\_\_ (self,pn,tt,aut):

super(). \_\_init\_\_(pn)

self.title=tt

self.author=aut

def display(self):

print("Title Name: ",self.title)

print("Author Name:",self.author)

class python(book):

def \_\_init\_\_ (self,pn,tt,aut,pr,pg):

super(). \_\_init\_\_(pn,tt,aut)

self.price=pr

self.page=pg

def pythondisplay(self):

print("Price: ",self.price)

print("No. of Pages: ",self.page)

obj=python("joy publishers","Python","Guido van Rossum",599,230);

obj.display()

obj.pythondisplay();

**output**

