CO4 PROGRAMS

**1.Create a Rectangular 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,breadth,length):

self.breadth=breadth

self.length=length

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())

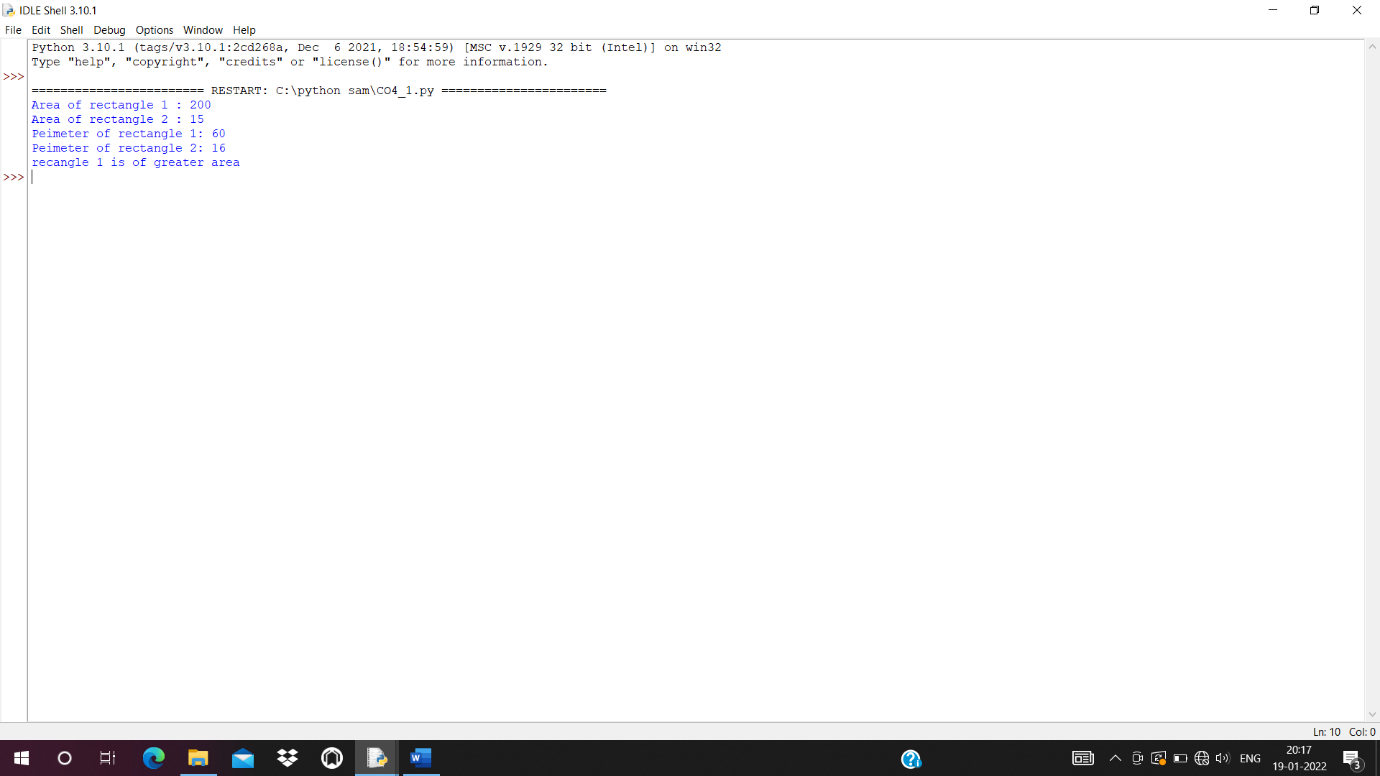
print("Peimeter of rectangle 2:",r2.perimeter())

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

print("recangle 1 is of greater area")

else:

print("recangle 2 is of greater area")



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print("Rs",w,"Successfully Withdrawn")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

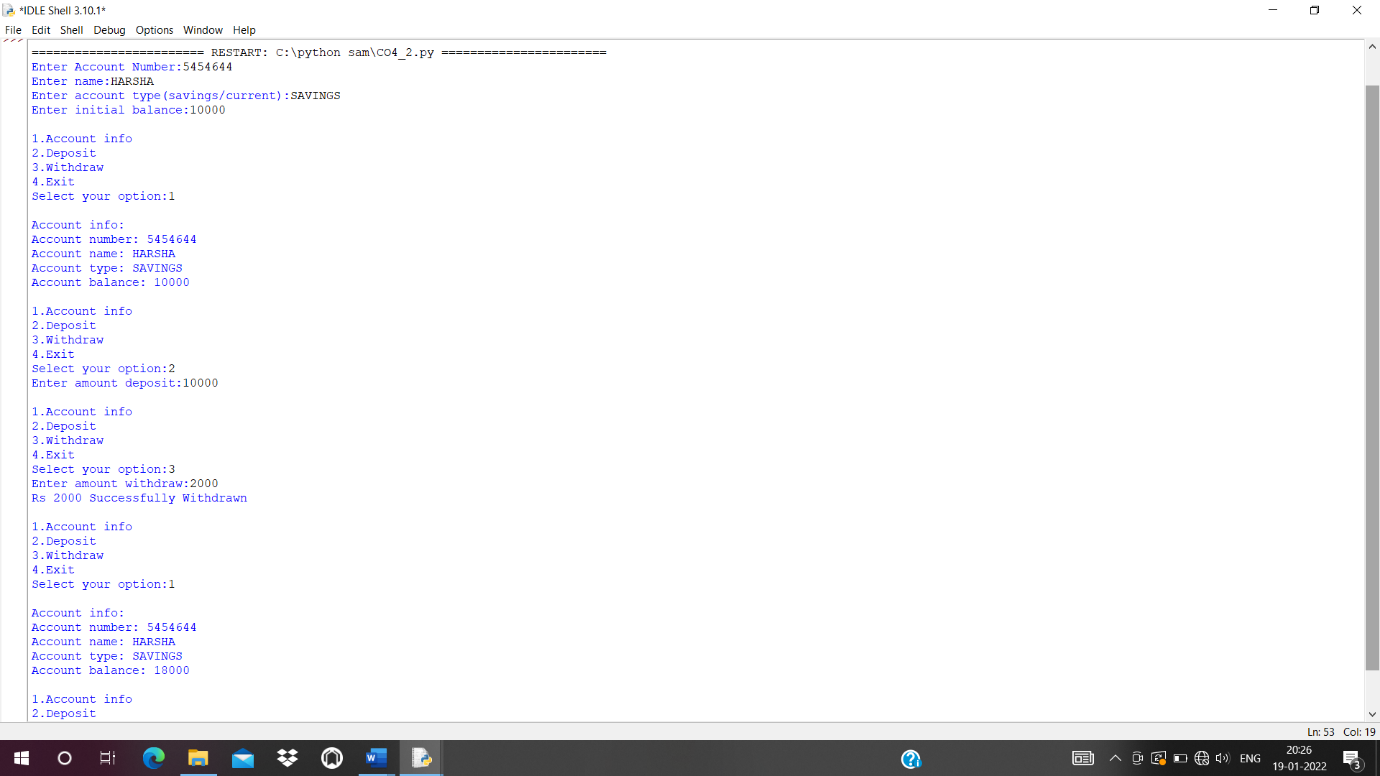
b1.withdraw()

elif opt == 4:

print("Exit")

break

else:

print("Invalid Option")

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

class rectangle:

\_\_area = 0

\_\_perimeter = 0

def \_\_init\_\_(self,length,width):

self.\_\_length = length

self.\_\_width = width

def calc\_area(self):

self.\_\_area = self.\_\_length\*self.\_\_width

print("Area is :",self.\_\_area)

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

if self.\_\_area < second.\_\_area:

return True

else:

return False

length1= int(input("Enter length of the rectangle 1 : "))

width1 = int(input("Enter width of the rectangle 1 : "))

length2 = int(input("Enter length of the rectangle 2 : "))

width2 = int(input("Enter width of the rectangle 2 : "))

obj1 = rectangle(length1,width1)

obj2 = rectangle(length2,width2)

obj1.calc\_area()

obj2.calc\_area()

if obj1 < obj2:

print("Rectangle two is large")

else:

print("Rectangle one is large or these are equal")



**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"))

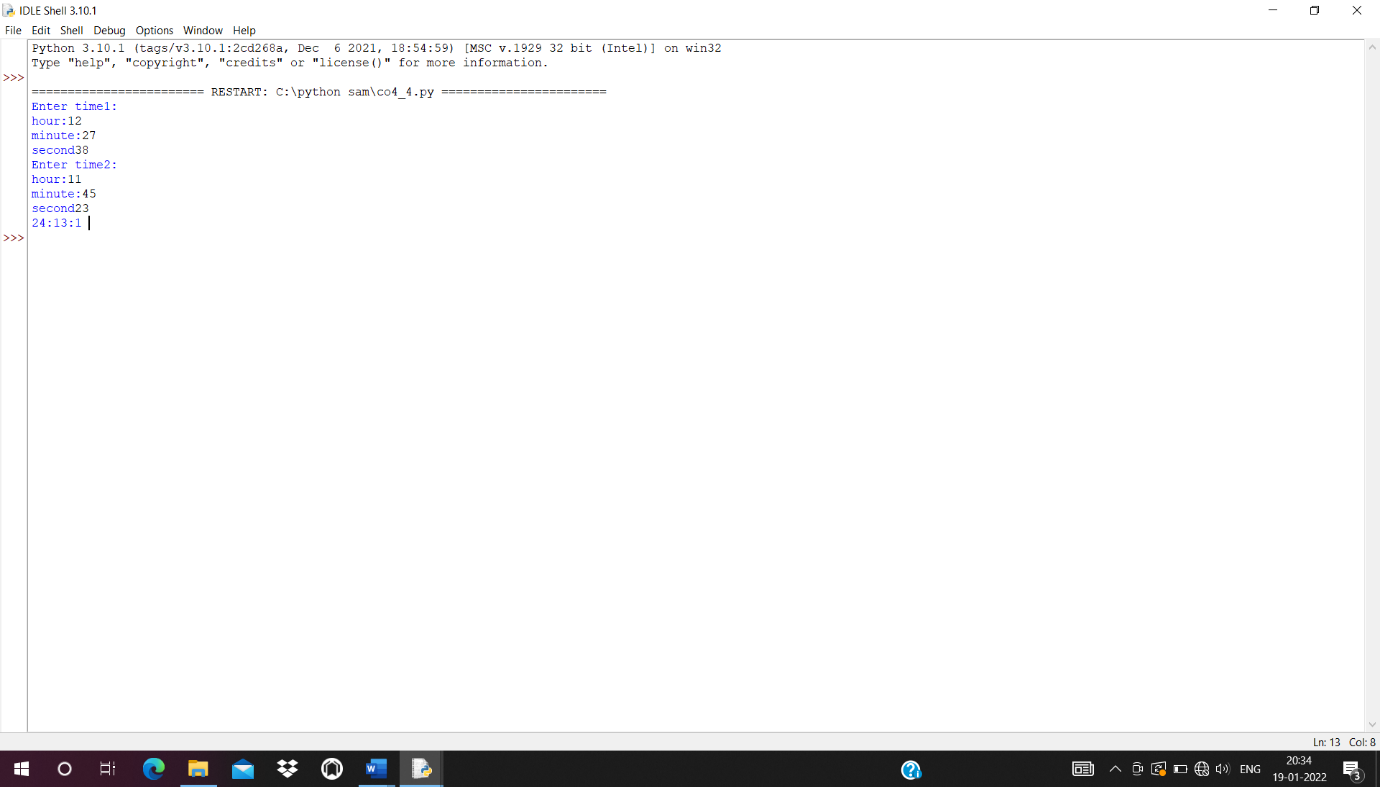
t2=Time(h2,m2,s2)

hr,min,sec=t1+t2

print(hr,end=":")

print(min,end=":")

print(sec,end=" ")



**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","ezhuthachan",1000,1000)

p.display()

