***CO-4 PYTHON LAB RECORD***

1. ***Pgm 1***

**Create Rectangle class with attributes length and breadth and methods to find area and**

**perimeter. Compare two Rectangle objects by their area.**

***Input***

class Rectangle:

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

self.length = length

self.width = width

def compute\_area(self):

return self.length \* self.width

def compute\_perimeter(self):

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

l= float(input('Please Enter the Length of the Rectangle: '))

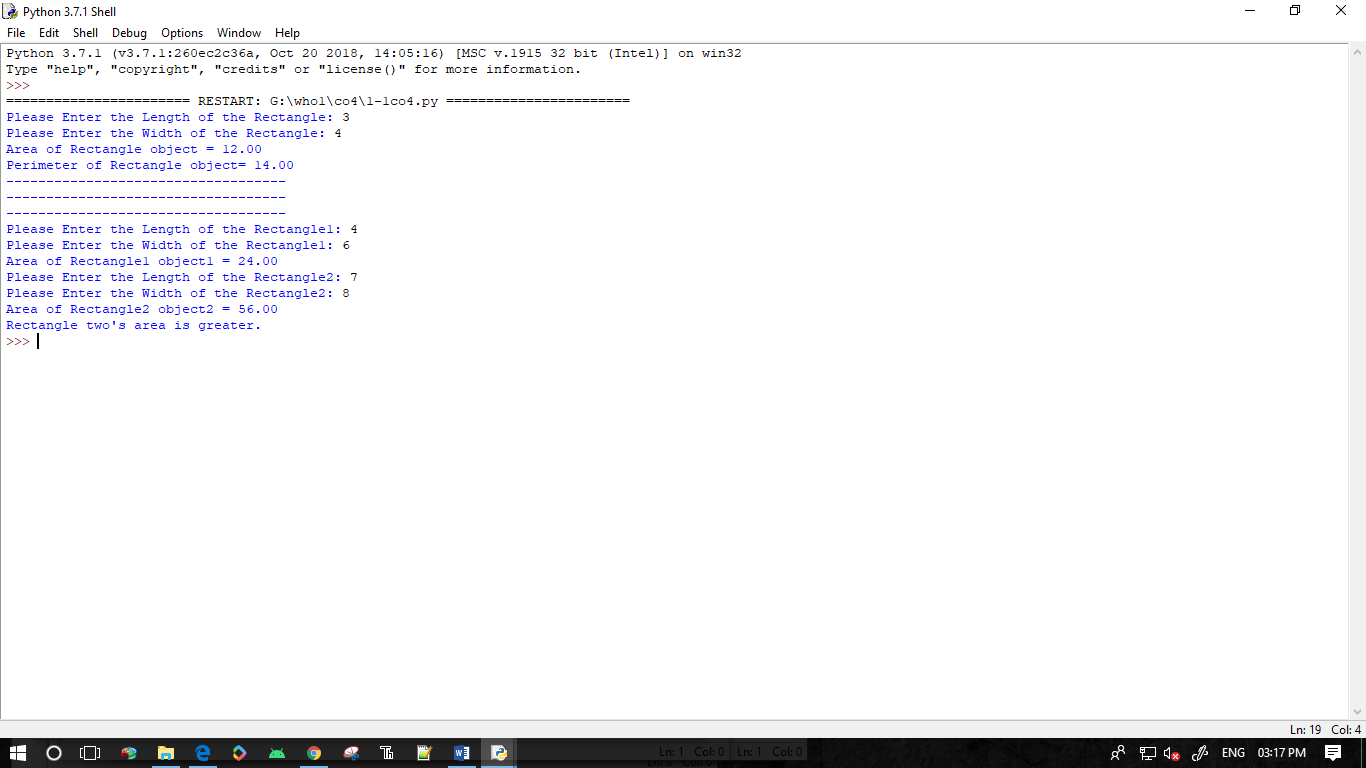
w= float(input('Please Enter the Width of the Rectangle: '))

object1 = Rectangle(l,w)

area = object1.compute\_area()

perimeter = object1.compute\_perimeter()

***out put***



1. ***Pgm 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.**

***Input:***

class bank:

\_\_acc\_name=""

\_\_acc\_no = ""

\_\_acc\_type = ""

\_\_acc\_balance = 0

def \_\_init\_\_(self,a\_name,a\_no,a\_type,a\_balance):

self.\_\_acc\_name = a\_name

self.\_\_acc\_no = a\_no

self.\_\_acc\_type = a\_type

self.\_\_acc\_balance = a\_balance

def deposite(self,a\_deposit):

print("Initial balance is : ",self.\_\_acc\_balance)

print("Deposite is : ",a\_deposit)

self.\_\_acc\_balance += a\_deposit

print("Current balance is : ",self.\_\_acc\_balance)

def withdraw(self):

print("Current balance is : ",self.\_\_acc\_balance)

self.amount = int(input("How much amount need to withdraw : "))

if self.amount > self.\_\_acc\_balance:

print("You don't have enough balance to withdraw !!")

print("Current balance is : ",self.\_\_acc\_balance)

else:

print(self.amount," is withrawed .")

self.\_\_acc\_balance -= self.amount

print("Current balance is : ",self.\_\_acc\_balance)

def acc\_info(self):

print("\n\n\n\n")

print("Account holder name : ",self.\_\_acc\_name)

print("Account number : ",self.\_\_acc\_no)

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

print("Account Balance is : ",self.\_\_acc\_balance)

print("\n")

def main():

name = input("Enter Account holder name : ")

no = input("Enter Account number : ")

atype = input("Enter Account type : ")

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

holder = bank(name,no,atype,bal)

while(True):

print("\n\n")

opt = int(input("1)Deposite \n2)Withdraw \n3)Account info \n0)Exit\nChoose your option :: "))

print("n\n")

if opt == 1:

amount = int(input("Deposite amount : "))

holder.deposite(amount)

elif opt == 2:

holder.withdraw()

elif opt == 3:

holder.acc\_info()

elif opt == 0:

break

else:

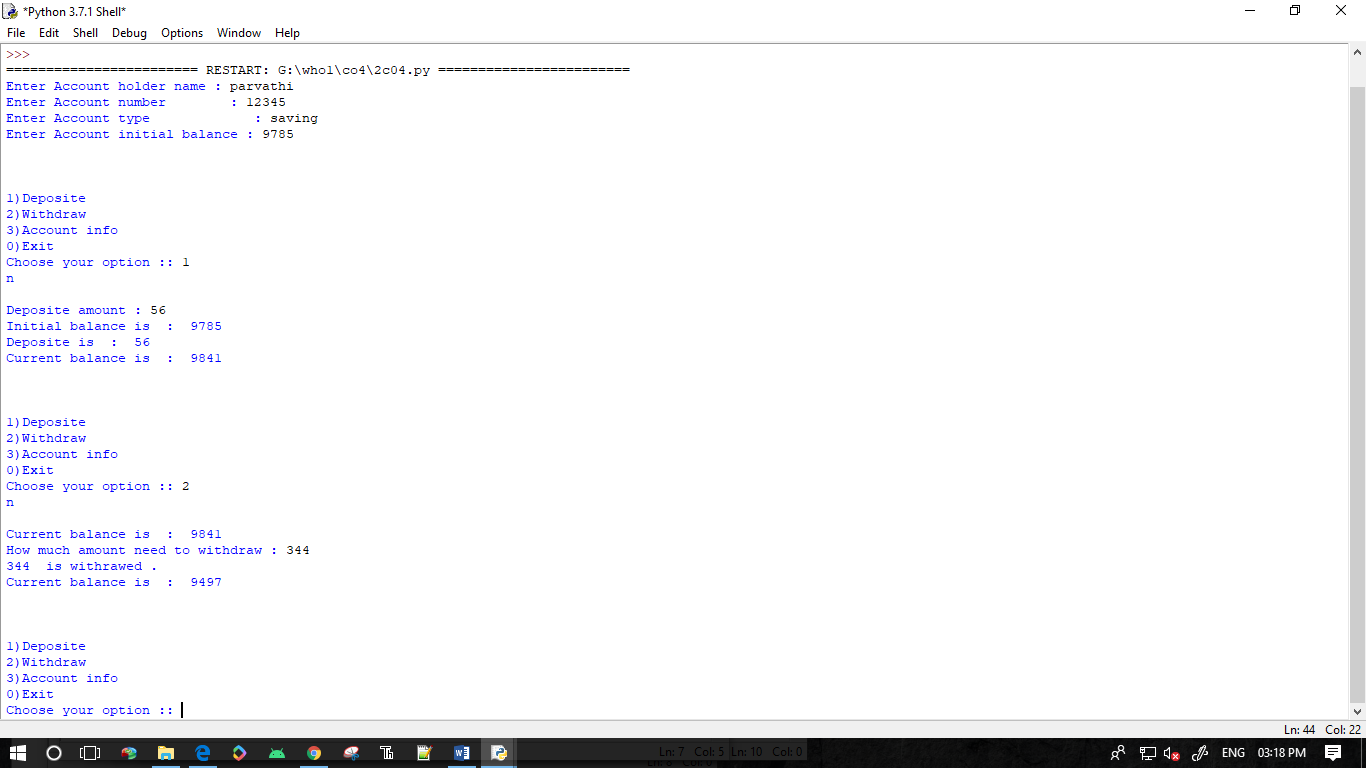
print("Invalid Option !")

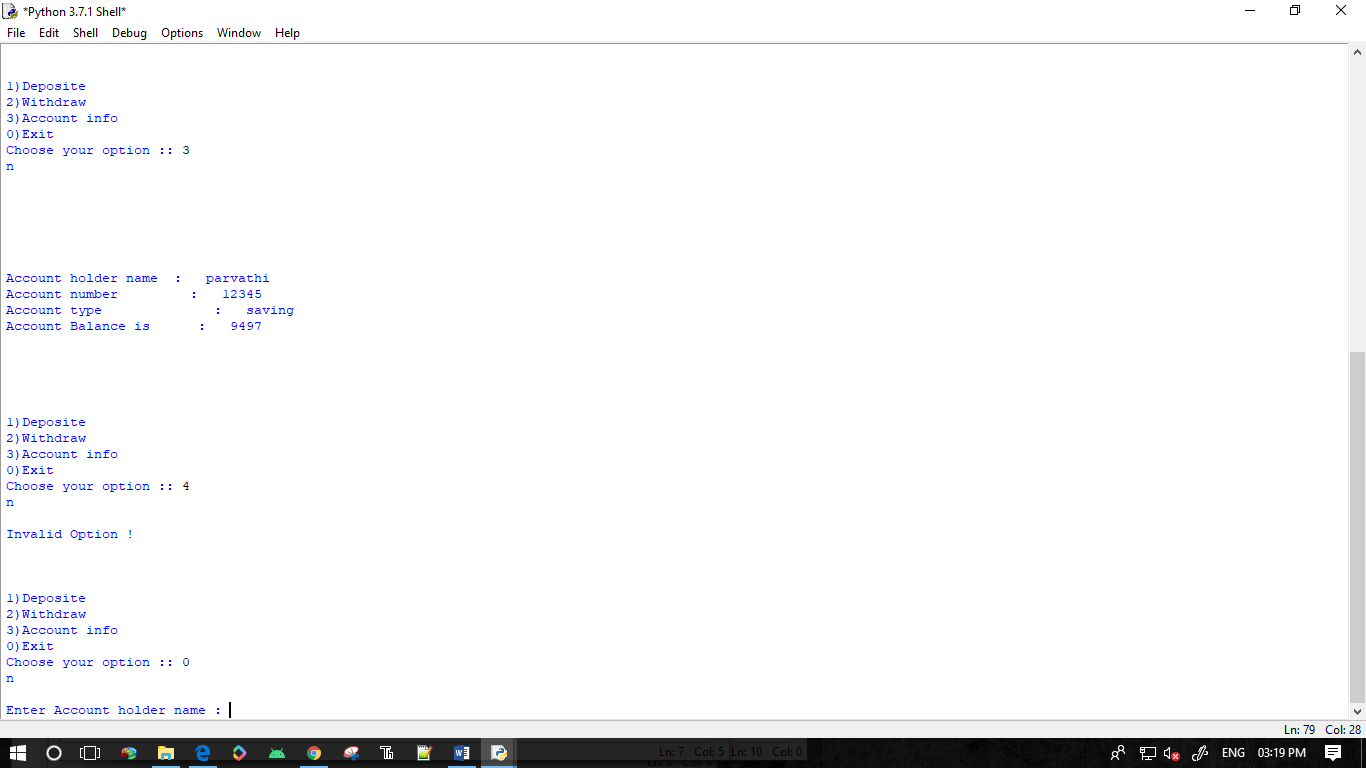
if \_\_name\_\_ == "\_\_main\_\_":

while(True):

main()

***out put***





1. ***Pgm 3***

**Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to**

**compare the area of 2 rectangles.**

***Input:***

class rectangle:

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

self.length=length

self.width=width

def \_\_lt\_\_(self,a1):

area1=self.length\*self.width

area2=a1.length\*a1.width

if(area1>area2):

return(True)

else:

return(False)

print("Enter the Details of Rectangle:1")

l1=int(input("Length:"))

w1=int(input("width:"))

r1=rectangle(l1,w1)

print("Enter the Details of Rectangle:2")

l2=int(input("Length:"))

w2=int(input("width:"))

r2=rectangle(l2,w2)

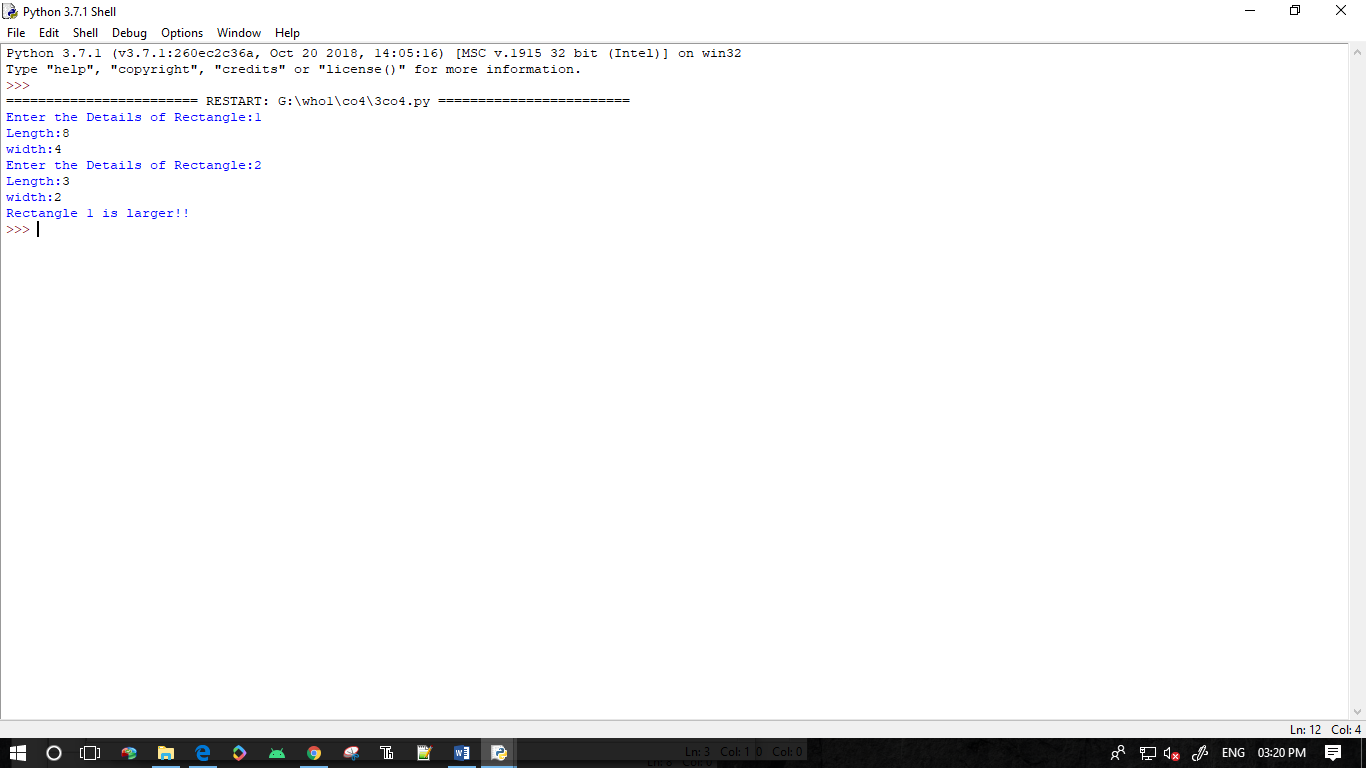
if(r1>r2):

print("Rectangle 2 is larger!!")

else:

print("Rectangle 1 is larger!!")

***out put***



1. ***Pgm 4***

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

**find sum of 2 time.**

***Input:***

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)

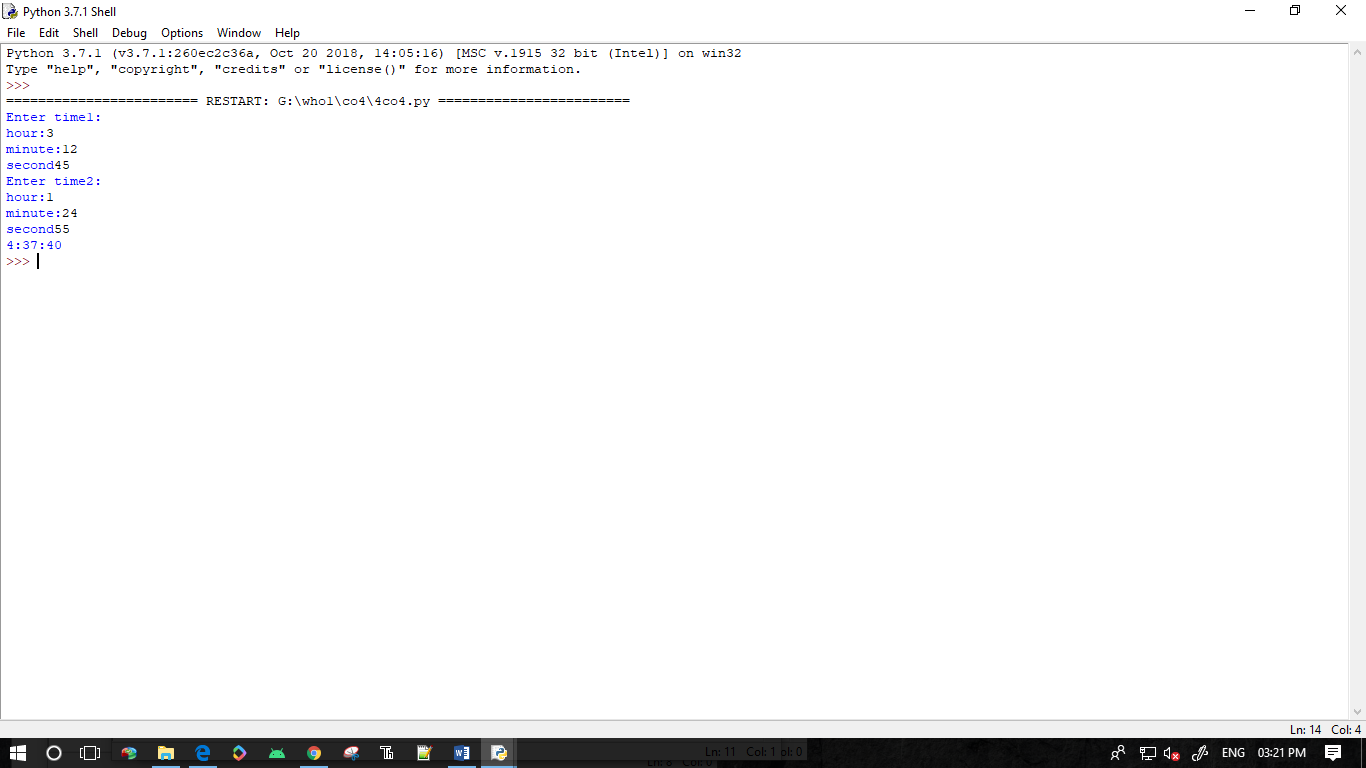
hr,min,sec=t1+t2

print(hr,end=":")

print(min,end=":")

print(sec,end=" ")

***out put***



1. ***Pgm 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.**

***Input:***

class publisher:

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

self.publishername=pn

def publisherdisplay(self):

print(self.publishername)

class book(publisher):

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

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

self.title=tt

self.author=aut

def bookdisplay(self):

print(self.title)

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

print("Title: ",self.title)

print("Author: ",self.author)

print("Price: ",self.price)

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

obj=python("Akshaya publishers","Python","Guido van Rossum",236,215);

obj.pythondisplay();

***Output:***

