

CO1:

2. Display future leap years from current year to a final year entered by user.

```
import datetime

a=datetime.datetime.now()

a=int(a.year)

b=int(input("Enter final year: "))

print("\nLeap Years:")

for i in range(a,b+1):

    if(i%4==0):

        print(i)
```

3. List comprehensions: (a) Generate positive list of numbers from a given list of integers

```
list1=[1,-1,2,-5,9,-2,-54,87,-33,-76,24,-67]

pos=list()

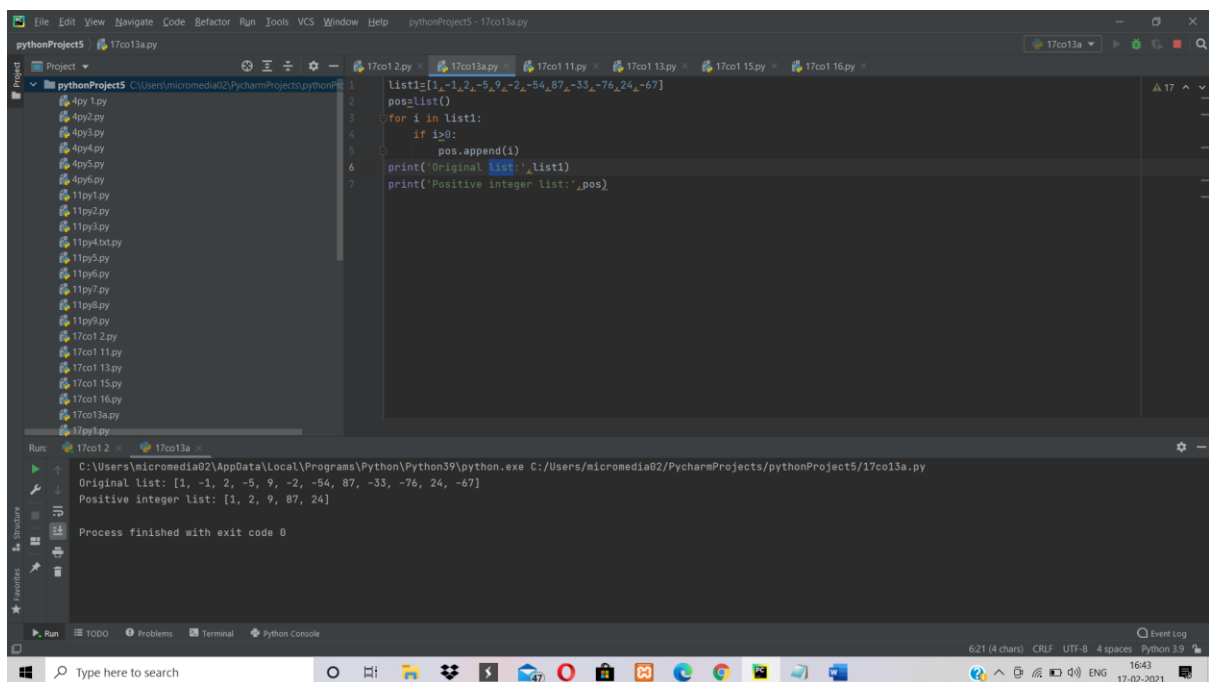
for i in list1:

    if i>0:

        pos.append(i)

print('Original list:',list1)

print('Positive integer list:',pos)
```



The screenshot shows the PyCharm IDE interface. The left sidebar displays a project tree with various files. The main editor window shows a Python script with the following code:

```
1 list1=[1,-1,2,-5,9,-2,-54,87,-33,-76,24,-67]
2 pos=list()
3 for i in list1:
4     if i>0:
5         pos.append(i)
6 print('Original list:',list1)
7 print('Positive integer list:',pos)
```

The bottom panel shows the Run output, indicating that the program executed successfully with exit code 0. The output is as follows:

```
Original list: [1, -1, 2, -5, 9, -2, -54, 87, -33, -76, 24, -67]
Positive integer list: [1, 2, 9, 87, 24]
```

The status bar at the bottom indicates the file encoding is UTF-8, the line ending is CRLF, and the Python version is 3.9.

11. Find biggest of 3 numbers entered.

```
a=int(input('Enter 1st no: '))
```

```
b=int(input('Enter 2nd no: '))
```

```
c=int(input('Enter 3rd no: '))
```

```
if a>b and b>c:
```

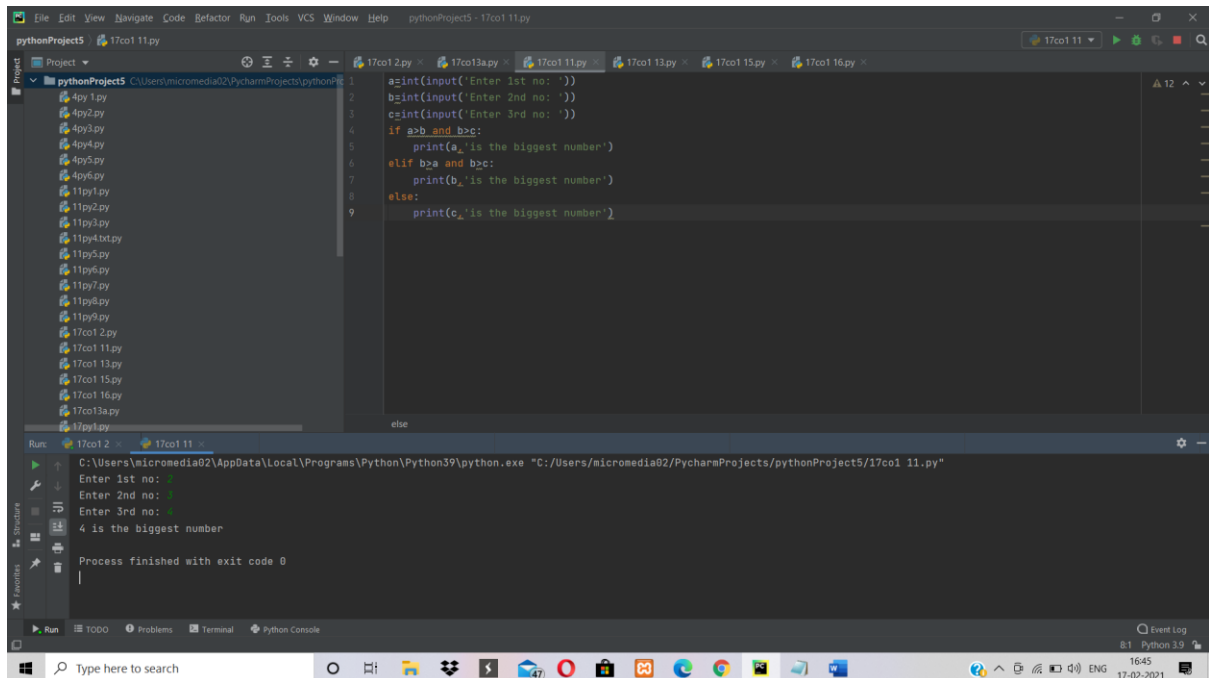
```
    print(a,'is the biggest number')
```

```
elif b>a and b>c:
```

```
    print(b,'is the biggest number')
```

```
else:
```

```
    print(c,'is the biggest number')
```

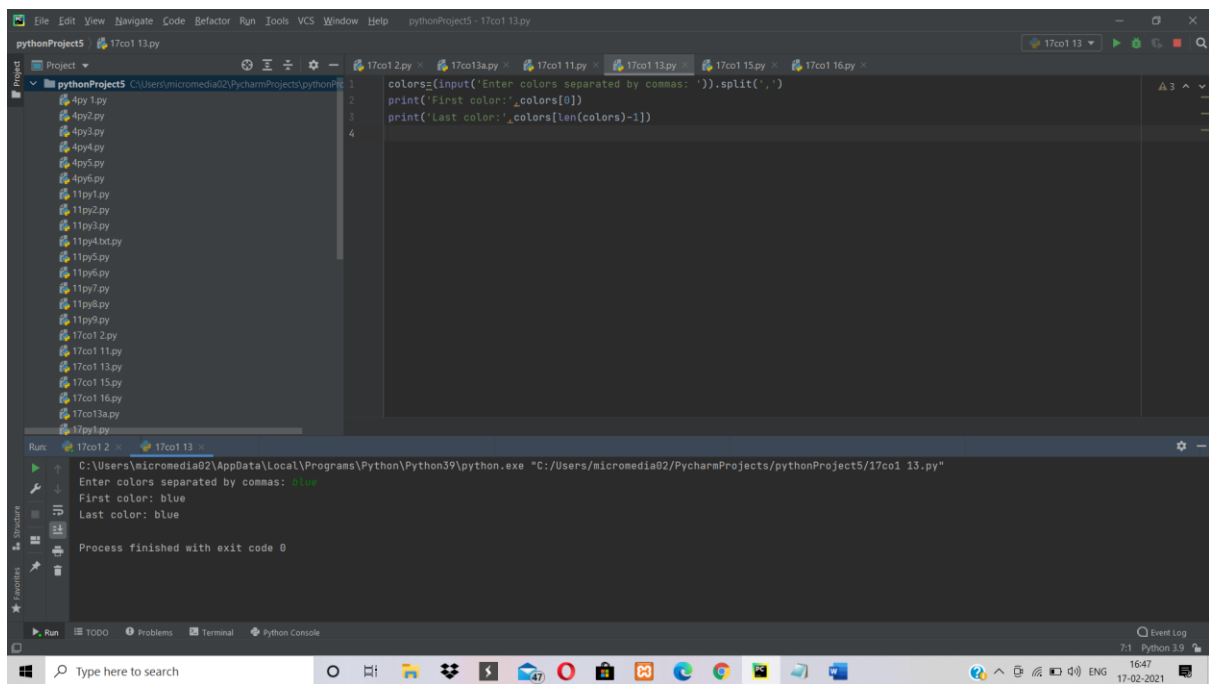


13. Create a list of colors from comma-separated color names entered by user. Display first and last colors.

```
colors=(input('Enter colors separated by commas:')).split(',')
```

```
print('First color:',colors[0])
```

```
print('Last color:',colors[len(colors)-1])
```

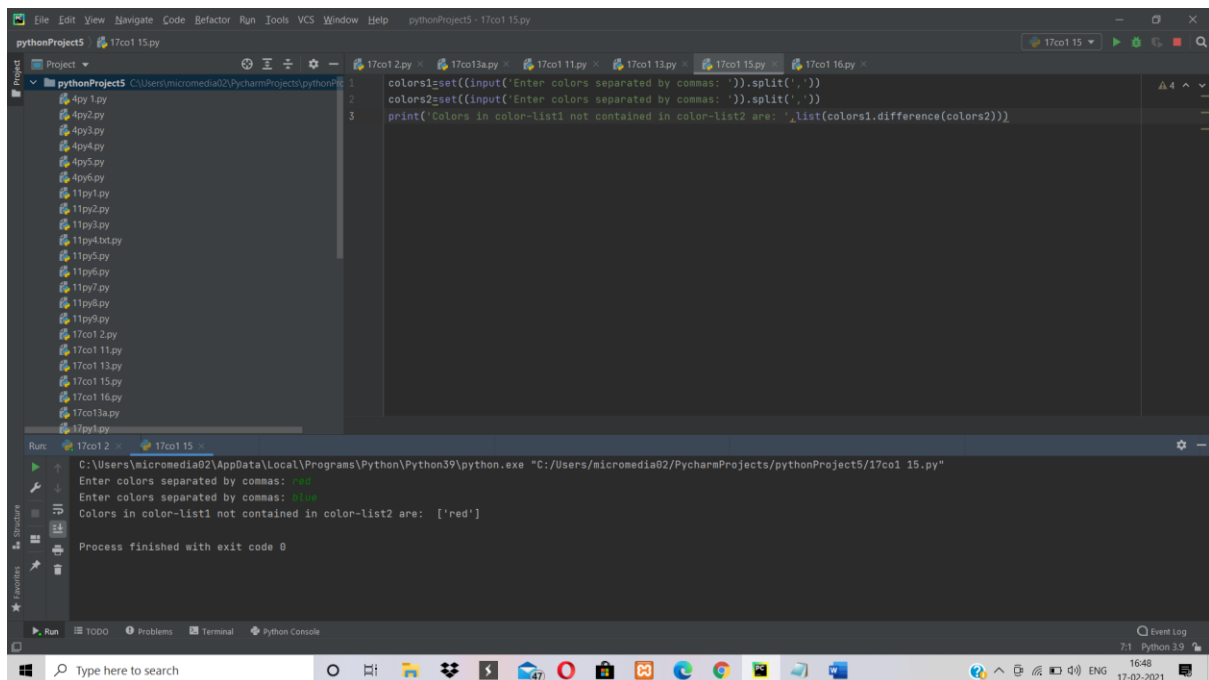


15. Print out all colors from color-list1 not contained in color-list2.

```
colors1=set((input('Enter colors separated by commas: ')).split(','))
```

```
colors2=set((input('Enter colors separated by commas: ')).split(','))
```

```
print('Colors in color-list1 not contained in color-list2 are: ',list(colors1.difference(colors2)))
```



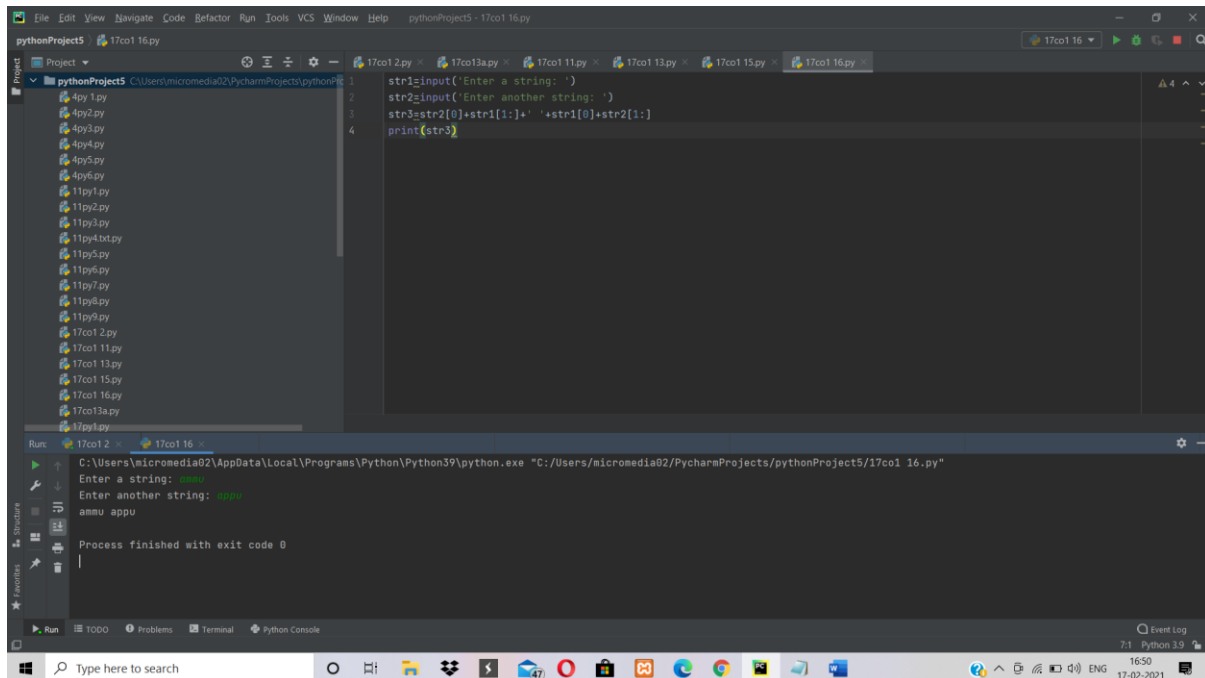
16. Create a single string separated with space from two strings by swapping the character at position 1.

```
str1=input('Enter a string: ')
```

```
str2=input('Enter another string: ')
```

```
str3=str2[0]+str1[1:]+ ' '+str1[0]+str2[1:]
```

```
print(str3)
```



CO3:

2. Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

Circle

```
def area(r):
```

```
    print('Area of circle with radius',r,'is:', '%.2f'%(3.14*r*r),'Sq.units')
```

```
def circumference(r):
```

```
    print('Circumference of circle with radius',r,'is:', '%.2f'%(3.14*2*r),'units')
```

```
area
```

```
import circle
```

```
from rectangle import *
```

```
from Graphics._3D_graphics import cuboid,sphere
```

```
a=float(input('Enter length of the rectangle: '))
```

```
b=float(input('Enter breadth of the rectangle: '))
```

```
area(a,b)
```

```
r=float(input('Enter the radius of the circle: '))
```

```

circle.area(r)

l=float(input('Enter length of the cuboid: '))
b=float(input('Enter breadth of the cuboid: '))
h=float(input('Enter height of the cuboid: '))
cuboid.area(l,b,h)

r=float(input('Enter the radius of the sphere: '))
sphere.area(r)

perimeter

import circle

from rectangle import *

from Graphics._3D_graphics import cuboid,sphere

a=float(input('Enter length of the rectangle: '))
b=float(input('Enter breadth of the rectangle: '))
perimeter(a,b)

r=float(input('Enter the radius of the circle: '))
circle.circumference(r)

l=float(input('Enter length of the cuboid: '))
b=float(input('Enter breadth of the cuboid: '))
h=float(input('Enter height of the cuboid: '))
cuboid.perimeter(l,b,h)

r=float(input('Enter the radius of the sphere: '))
sphere.perimeter(r)

import circle

from rectangle import *

from Graphics._3D_graphics import cuboid,sphere

a=float(input('Enter length of the rectangle: '))
b=float(input('Enter breadth of the rectangle: '))
perimeter(a,b)

r=float(input('Enter the radius of the circle: '))
circle.circumference(r)

l=float(input('Enter length of the cuboid: '))
b=float(input('Enter breadth of the cuboid: '))

```

```

h=float(input('Enter height of the cuboid: '))
cuboid.perimeter(l,b,h)
r=float(input('Enter the radius of the sphere: '))
sphere.perimeter(r)
rectangle
def area(a,b):
    print('Area of rectangle with sides',a,'and',b,'is: ', '%.2f'%(a*b),'Sq.units')
def perimeter(a,b):
    print('Perimeter of rectangle with sides',a,'and',b,'is:', '%.2f'%(2*(a+b)),'units')
3d graphics
Cuboid
def area(l,b,h):
    print('Total surface area of cuboid with
dimensions',l,',',b,',',h,'is:', '%.2f'%(2*((l*b)+(b*h)+(l*h))), 'Sq.units')
def perimeter(l,b,h):
    print('Perimeter of cuboid with dimensions', l, ',', b, ',', h, 'is:', '%.2f%(4*(l+b+h))','units')
sphere
def area(r):
    print('Area of sphere with radius',r,'is:', '%.2f%(4*(3.14*r*r))','Sq.units')
def perimeter(r):
    print('Perimeter of (great circle of) sphere with radius',r,'is:', '%.2f%(2*3.14*r)','units')

```

```

Perimeter of a circle with radius 10 is  62.83185307179586
Area of a circle with radius 10 is :  314.1592653589793
Area of a Rectangle with length and width 10 is :  100
Perimeter of a Rectangle with length and width 10 is :  40
Area of a cuboid with length,width,height 10 is :  600
Perimeter of a cuboid with length,width,height 10 is :  120
Area of a sphere with radius 10 is :  1256.6370614359173
Perimeter of a sphere with radius 10 is  62.83185307179586

```

Course Outcome 4 (CO4):

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.length=l

    self.breadth=b

    def area(self):

return self.length*self.breadth

    def perimeter(self):

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

    def cmp(self,obj):
        if self.area()>obj.area():
            print('Rectangle with length =',self.length,'and breadth =',self.breadth,'has the greater area')
        elif self.area()<obj.area():

print('Rectangle with length =',obj.length, 'and breadth =', obj.breadth, 'has the greater area')

        else:
            print('They have equal area!')
r1=Rectangle(9,3)

r2=Rectangle(3,4)

r1.cmp(r2)

```

```

class Rectangle:
    def __init__(self,l,b):
        self.length=l
        self.breadth=b
    def area(self):
        return self.length*self.breadth
    def perimeter(self):
        return 2*(self.length+self.breadth)
    def cmp(self,obj):
        if self.area()>obj.area():
            print('Rectangle with length =',self.length,'and breadth =',self.breadth,'has the greater area')
        elif self.area()<obj.area():
            print('Rectangle with length =',obj.length, 'and breadth =', obj.breadth, 'has the greater area')
        else:
            print('They have equal area!')

r1=Rectangle(9,3)
r2=Rectangle(3,4)
r1.cmp(r2)

```

Run: 17co1 2 - 17co41

C:\Users\micromedia02\AppData\Local\Programs\Python\Python39\python.exe C:\Users\micromedia02\PycharmProjects\pythonProject5\17co41.py

Rectangle with length = 9 and breadth = 3 has the greater area

Process finished with exit code 0

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 BankAccount:

    def __init__(self,a,n,t,b):

```

```
self.acno=a

self.name=n

self.type=t

self.bal=b

def deposit(self,a):

    self.bal+=a

    print('Rs.',a,'deposited! Current balance is: Rs.',self.bal)

def withdraw(self,a):

    if self.bal >= a:

        self.bal-=a

        print('Rs.',a,'withdrawn! Current balance is: Rs.', self.bal)

    else:

        print('Insufficient balance to make this transaction!')

a=int(input('Enter account number:'))

n=input('Enter name of the account holder: ')

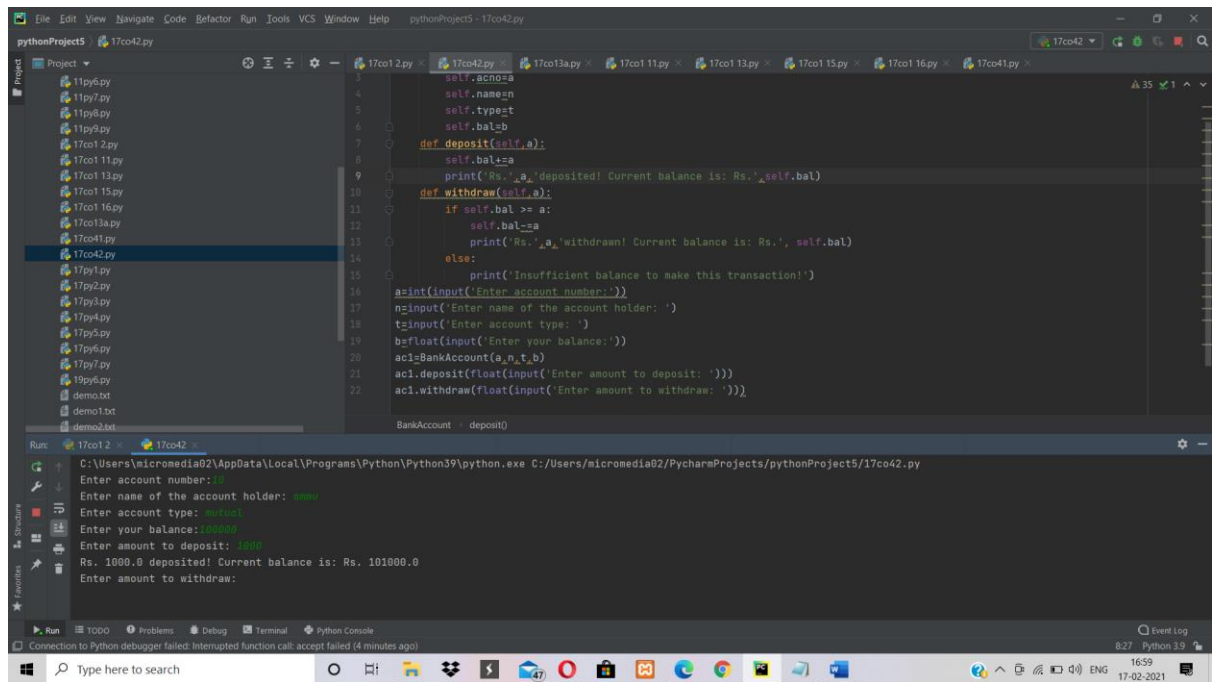
t=input('Enter account type: ')

b=float(input('Enter your balance:'))

ac1=BankAccount(a,n,t,b)

ac1.deposit(float(input('Enter amount to deposit: ')))

ac1.withdraw(float(input('Enter amount to withdraw: ')))
```

3. Create a class Rectangle with private attributes length and width. Overload ‘

class Rectangle:

```
def __init__(self,l,w):
```

```
    self.__length = l
```

```
    self.__width = w
```

```
    self.area=self.__width * self.__length
```

```
def __lt__(self, other):
```

```
    if self.area < other.area:
```

```
        print('Rectangle with length=',self.__length,'and width=',self.__width,'has the lesser area!')
```

```
    elif other.area < self.area:
```

```
        print('Rectangle with length=',other.__length,'and width=',other.__width,'has the lesser area!')
```

```
    else:
```

```
        print('They have equal area!')
```

```
l=float(input('Enter length of 1st rectangle: '))
```

```
w=float(input('Enter width of 1st rectangle: '))
```

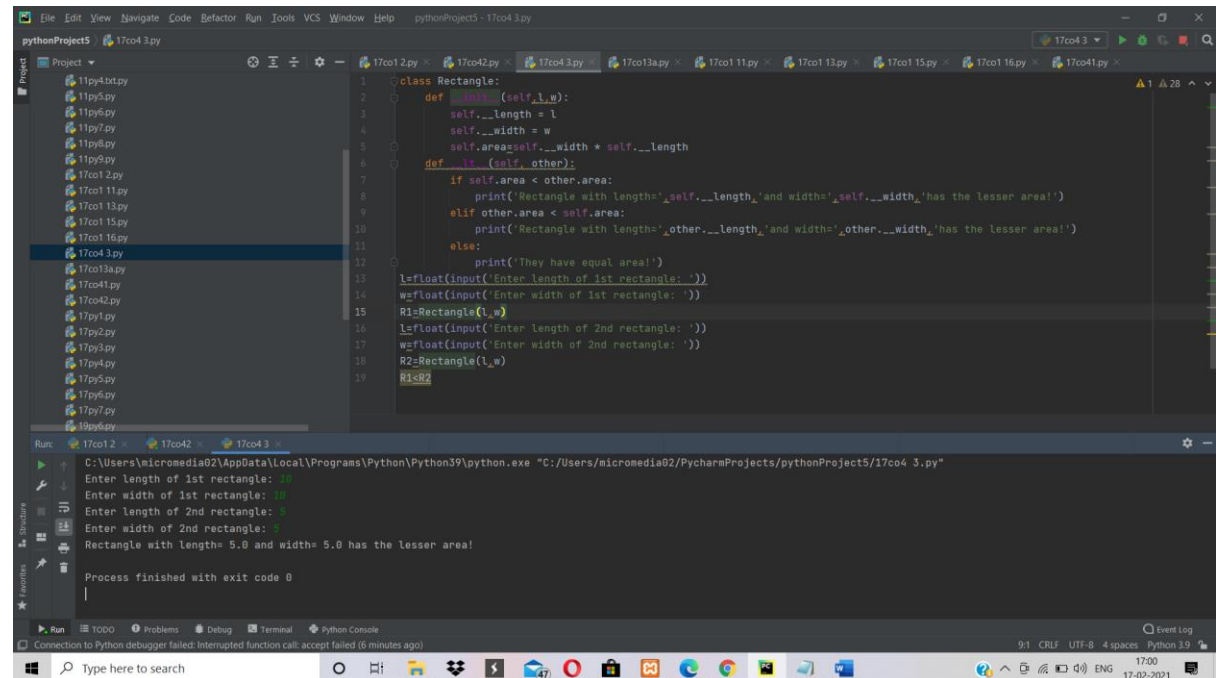
```
R1=Rectangle(l,w)
```

```
l=float(input('Enter length of 2nd rectangle: '))
```

```
w=float(input('Enter width of 2nd rectangle: '))
```

R2=Rectangle(l,w)

R1<R2



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,hh=0,mm=0,ss=0):
```

```
    self.__hour=hh
```

```
    self.__minute=mm
```

```
    self.__second=ss
```

```
def __add__(self,other):
```

```
    second=int((self.__second + other.__second)%60)
```

```
    minute=int((self.__minute + other.__minute)%60 + ((self.__second + other.__second)/60))
```

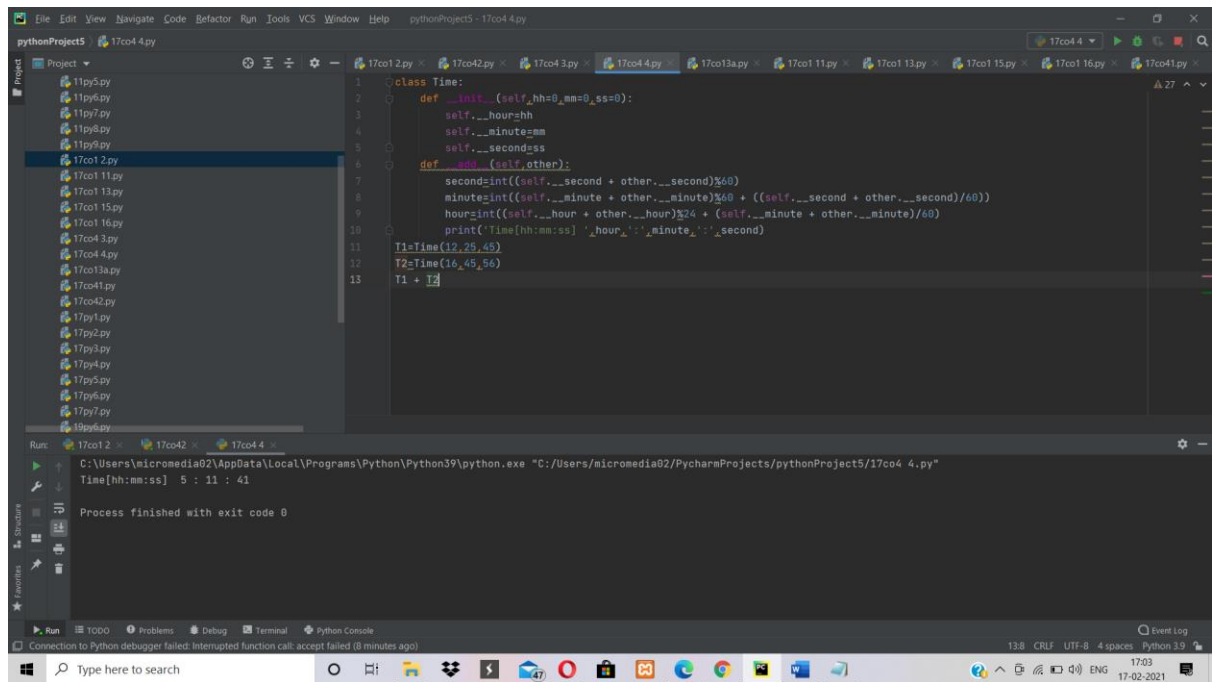
```
    hour=int((self.__hour + other.__hour)%24 + (self.__minute + other.__minute)/60)
```

```
    print('Time[hh:mm:ss] ',hour,':',minute,':',second)
```

T1=Time(12,25,45)

T2=Time(16,45,56)

T1 + T2



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, name1):
    self.name = name1

def show(self):
    pass
```

class Book(Publisher):

```
def __init__(self, title1, author1, name1):
    self.title = title1
    self.author = author1
    Publisher.__init__(self, name1)

def show(self):
    pass
```

class Python(Book):

```
def __init__(self, p, no, title1, author1, name1):
    self.price = p
    self.no_of_pages = no
```

```
Book.__init__(self,title1,author1,name1)
```

```
def show(self):
```

```
    print('Book title:',self.title)
```

```
    print('Author:',self.author)
```

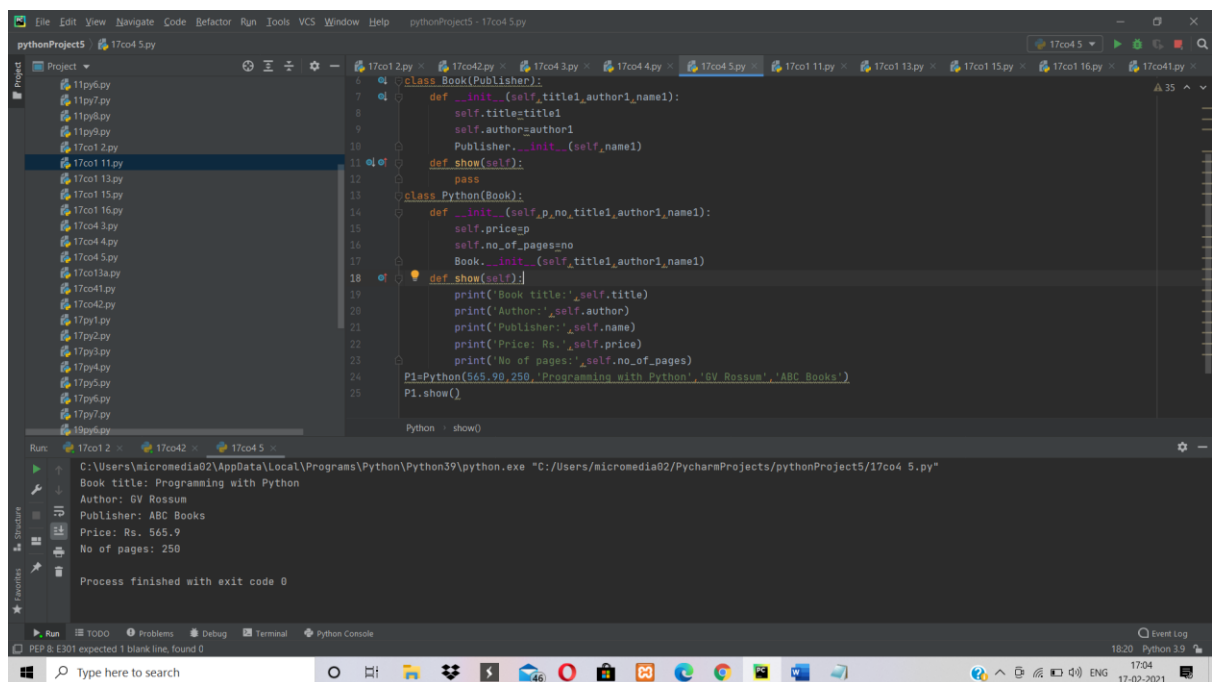
```
    print('Publisher:',self.name)
```

```
    print('Price: Rs.',self.price)
```

```
    print('No of pages:',self.no_of_pages)
```

```
P1=Python(565.90,250,'Programming with Python','GV Rossum','ABC Books')
```

```
P1.show()
```



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script with the following code:

```
class Book(Publisher):
    def __init__(self,title1,author1,name1):
        self.title=title1
        self.author=author1
        Publisher.__init__(self,name1)
    def show(self):
        pass
class Python(Book):
    def __init__(self,p,no,title1,author1,name1):
        self.price=p
        self.no_of_pages=no
        Book.__init__(self,title1,author1,name1)
    def show(self):
        print('Book title:',self.title)
        print('Author:',self.author)
        print('Publisher:',self.name)
        print('Price: Rs.',self.price)
        print('No of pages:',self.no_of_pages)

P1=Python(565.90,250,'Programming with Python','GV Rossum','ABC Books')
P1.show()
```

The Run window at the bottom shows the output of the program:

```
C:\Users\micromedia02\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/micromedia02/PycharmProjects/pythonProject5/17co4 5.py"
Book title: Programming with Python
Author: GV Rossum
Publisher: ABC Books
Price: Rs. 565.9
No of pages: 250
Process finished with exit code 0
```