

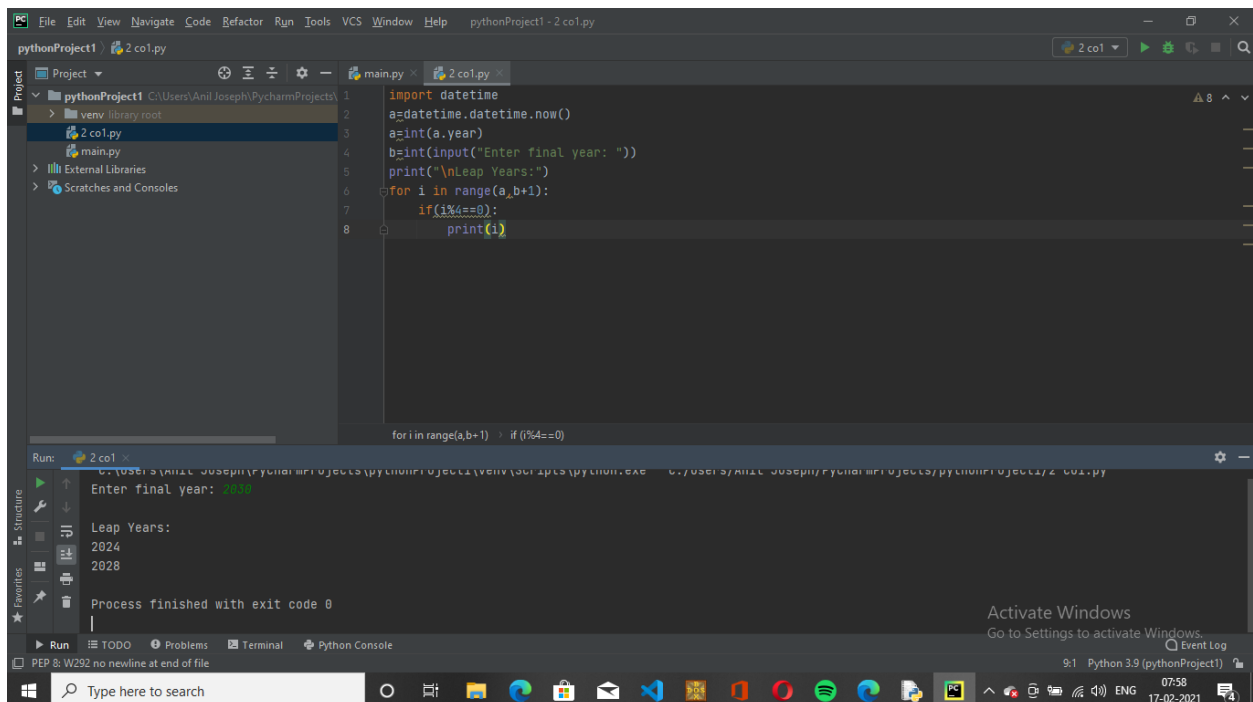
17/2/2021

Course Outcome 1 (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)
```

Output



The screenshot shows a PyCharm IDE with a Python project named 'pythonProject1'. The file '2 co1.py' is open, containing the following code:

```
1 import datetime  
2 a=datetime.datetime.now()  
3 a=int(a.year)  
4 b=int(input("Enter final year: "))  
5 print("\nLeap Years:")  
6 for i in range(a,b+1):  
7     if(i%4==0):  
8         print(i)
```

The Run console at the bottom shows the execution output:

```
Enter final year: 2030  
Leap Years:  
2024  
2028  
Process finished with exit code 0
```

The status bar at the bottom indicates 'Python 3.9 (pythonProject1)' and the date '17-02-2021'.

3. List comprehensions:

(a) Generate positive list of numbers from a given list of integers

```
list1=[56,25,85,-44,-620-95,55-50,66]
```

```
pos=list()
```

```
for i in list1:
```

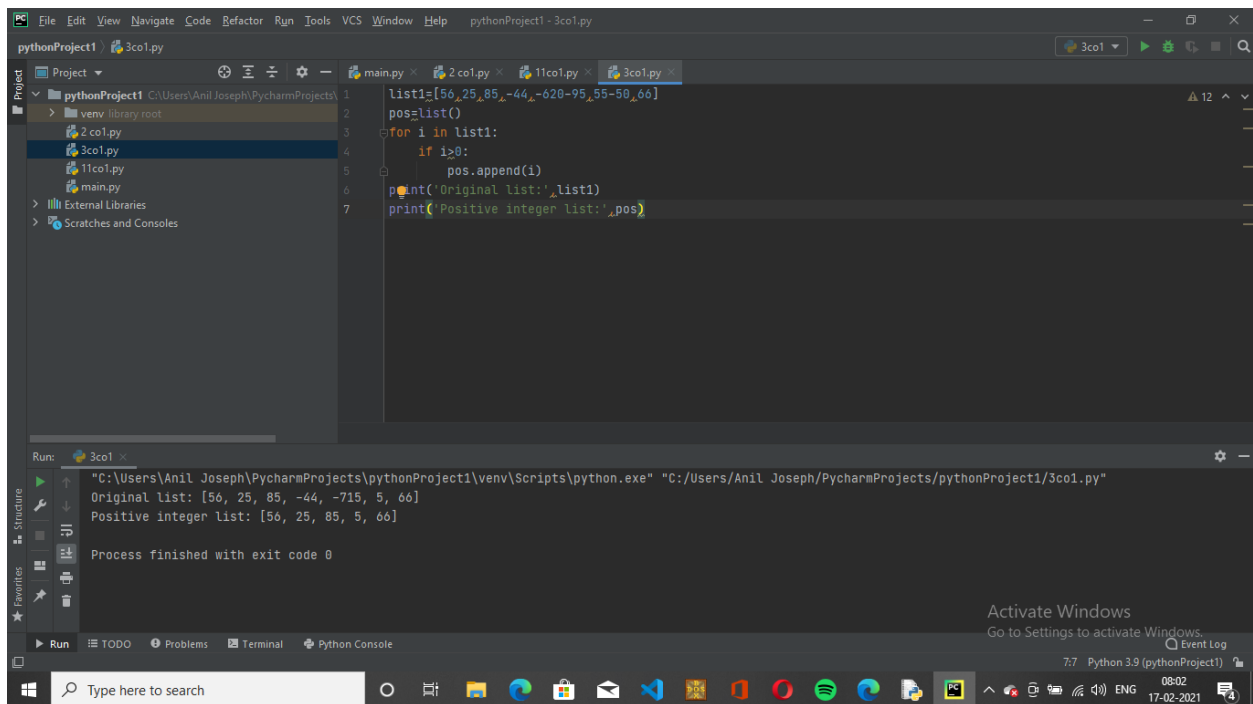
```
    if i>0:
```

```
        pos.append(i)
```

```
print('Original list:',list1)
```

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

Output



```
pythonProject1 - 3co1.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1 - 3co1.py
pythonProject1 C:\Users\Anil Joseph\PycharmProjects\
venv library root
2co1.py
3co1.py
1co1.py
main.py
External Libraries
Scratches and Consoles
main.py x 2co1.py x 1co1.py x 3co1.py x
1 list1=[56,25,85,-44,-620-95,55-50,66]
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)

Run: 3co1 x
"C:\Users\Anil Joseph\PycharmProjects\pythonProject1\venv\Scripts\python.exe" "C:/Users/Anil Joseph/PycharmProjects/pythonProject1/3co1.py"
Original list: [56, 25, 85, -44, -718, 5, 66]
Positive integer list: [56, 25, 85, 5, 66]
Process finished with exit code 0

Activate Windows
Go to Settings to activate Windows.
7:7 Python 3.9 (pythonProject1)
Type here to search 08:02 17-02-2021
```

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')
```

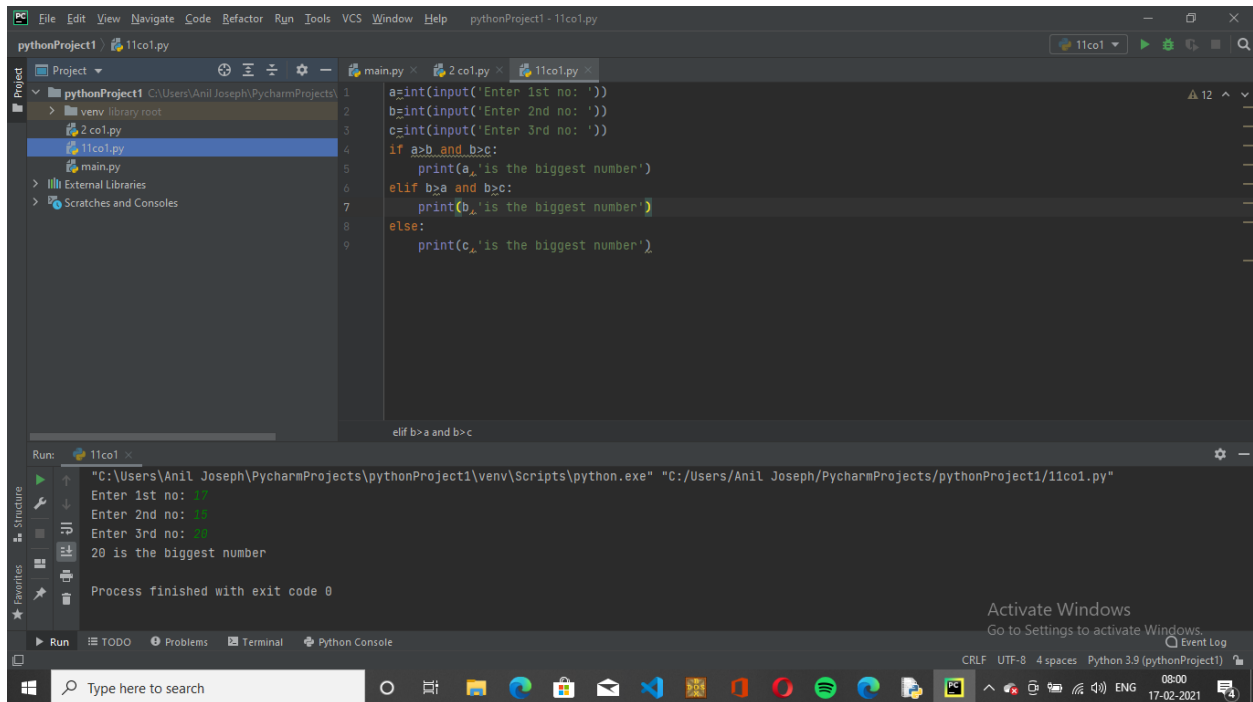
```
elif b>a and b>c:
```

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

else:

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

Output



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `11co1.py` with the following code:

```
1 a=int(input('Enter 1st no: '))
2 b=int(input('Enter 2nd no: '))
3 c=int(input('Enter 3rd no: '))
4 if a>b and b>c:
5     print(a,'is the biggest number')
6 elif b>a and b>c:
7     print(b,'is the biggest number')
8 else:
9     print(c,'is the biggest number')
```

The left sidebar shows the project structure with `11co1.py` selected. The bottom panel shows the Run console output:

```
Run: 11co1
"C:\Users\Anil Joseph\PycharmProjects\pythonProject1\venv\Scripts\python.exe" "C:/Users/Anil Joseph/PycharmProjects/pythonProject1/11co1.py"
Enter 1st no: 12
Enter 2nd no: 13
Enter 3rd no: 20
20 is the biggest number
Process finished with exit code 0
```

The Windows taskbar at the bottom shows the system clock as 08:00 on 17-02-2021.

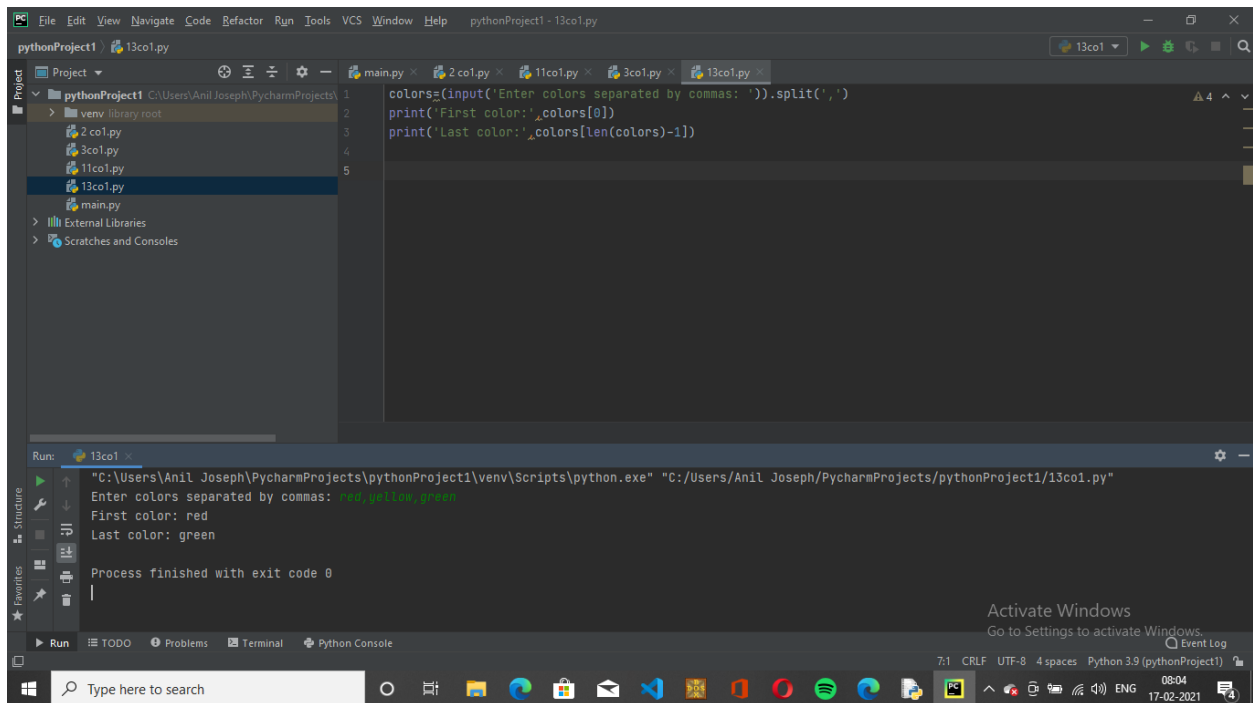
13. Create a list of colors from comma-separated color names entered by user. Display first and last colors.

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

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

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

Output



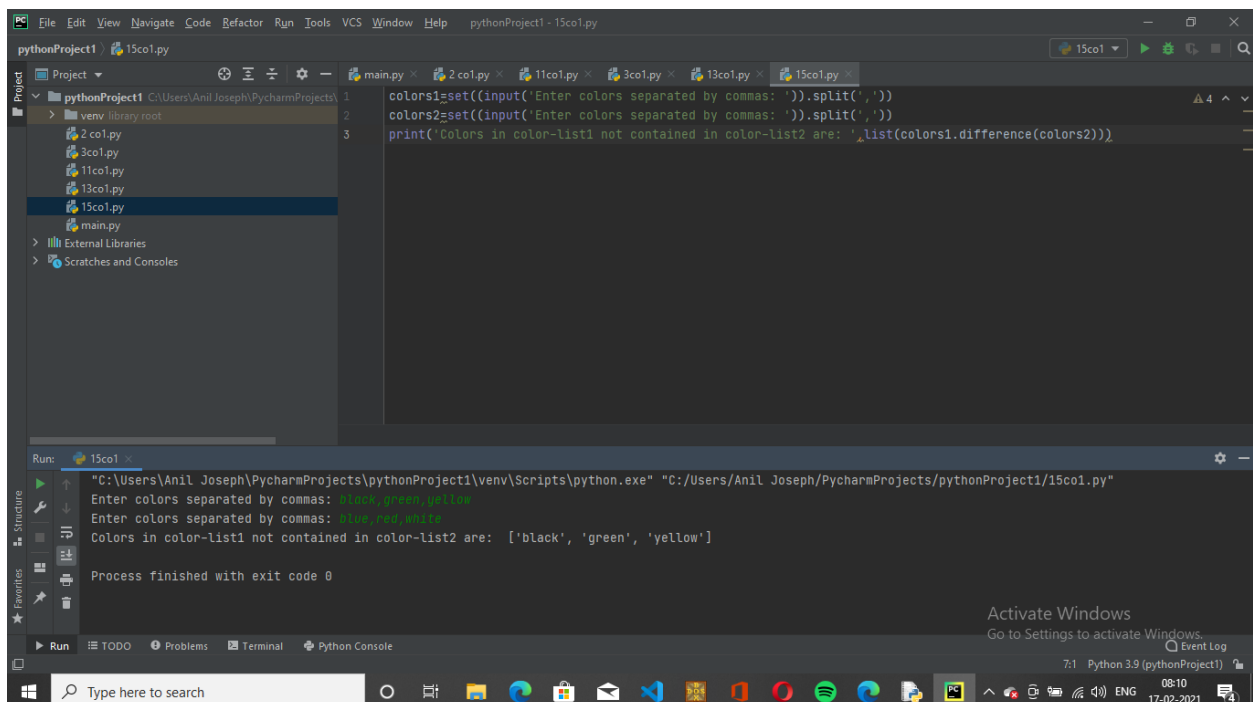
15. Print out all colors from color-list1 not contained in color-list2.

```
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)))
```

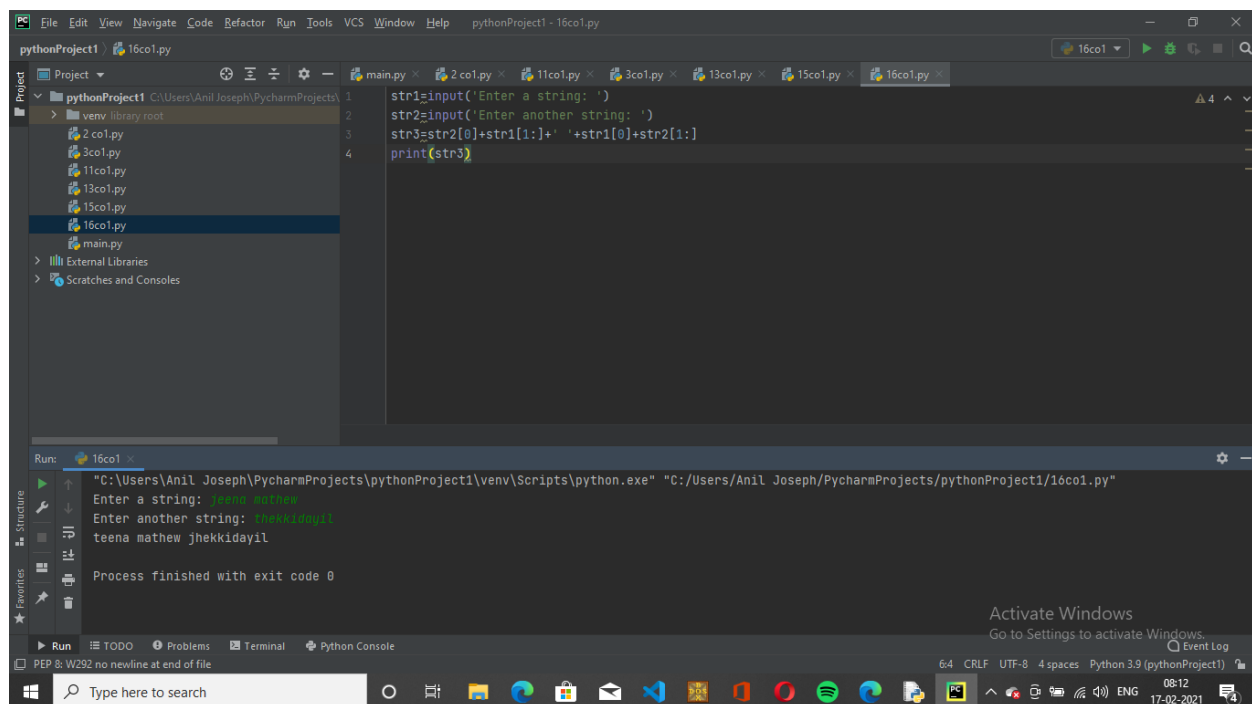
Output



16. Create a single string separated with space from two strings by swapping the

```
str1=input('Enter a string: ')
str2=input('Enter another string: ')
str3=str2[0]+str1[1:]+ ' '+str1[0]+str2[1:]
print(str3)
```

Output



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1 - 16co1.py
pythonProject1 16co1.py
Project
  pythonProject1 C:\Users\Anil Joseph\PycharmProjects\
    venv library root
    2co1.py
    3co1.py
    11co1.py
    13co1.py
    15co1.py
    16co1.py
    main.py
  External Libraries
  Scratches and Consoles
Run: 16co1
  "C:\Users\Anil Joseph\PycharmProjects\pythonProject1\venv\Scripts\python.exe" "C:/Users/Anil Joseph/PycharmProjects/pythonProject1/16co1.py"
  Enter a string: teena mathew
  Enter another string: jhekkidayil
  teena mathew jhekkidayil
  Process finished with exit code 0
Activate Windows
Go to Settings to activate Windows.
PEP 8: W292 no newline at end of file
Type here to search
```

Course Outcome 3(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.py

findarea.py

```
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)
```

findperimeter.py

```
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.py

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.py

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.py

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')

```

Output

```

Permeter 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
Permeter of a Rectangle with length and width 10 is :  40
Area of a  cuboid with length,width,height 10 is :  600
Permeter of a cuboid with length,width,height 10 is :  120
Area of a spere with radius 10 is :  1256.6370614359173
Permeter of a spere with radius 10 is  62.83185307179586

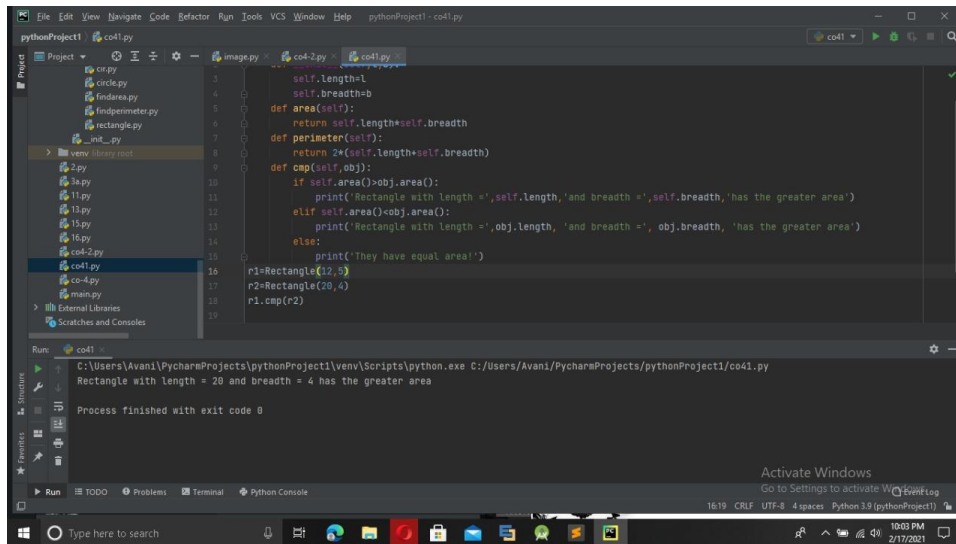
```

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(7,5)
r2=Rectangle(8,4)
r1.cmp(r2)
```

Output



```
pythonProject1 - co41.py
1 self.length=l
2 self.breadth=b
3
4 def area(self):
5     return self.length*self.breadth
6
7 def perimeter(self):
8     return 2*(self.length+self.breadth)
9
10 def cmp(self,obj):
11     if self.area()>obj.area():
12         print('Rectangle with length =',self.length,'and breadth =',self.breadth,'has the greater area')
13     elif self.area()<obj.area():
14         print('Rectangle with length =',obj.length, 'and breadth =', obj.breadth, 'has the greater area')
15     else:
16         print('They have equal area!')
17
18 r1=Rectangle(7,5)
19 r2=Rectangle(8,4)
20 r1.cmp(r2)
```

Run: co41

C:\Users\Avani\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/Avani/PycharmProjects/pythonProject1/co41.py

Rectangle with length = 8 and breadth = 4 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

```
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: ')))

```

## Output

The screenshot shows a Python IDE with a project named 'pythonProject1'. The file 'cod-2.py' is open, displaying the BankAccount class and its methods. The Run console shows the following output:

```

Enter account number: 526309989795
Enter name of the account holder: jessu mathew
Enter account type: joined ac
Enter your balance: 500
Enter amount to deposit: 800
Rs. 800.0 deposited! Current balance is: Rs. 1430.0
Enter amount to withdraw: 440
Rs. 500.0 withdrawn! Current balance is: Rs. 930.0

```

constructor and methods to deposit at the bank and withdraw an amount from the bank.

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,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
```

Output

Enter length of 1st rectangle: 5

Enter width of 1st rectangle: 3

Enter length of 2nd rectangle: 9

Enter width of 2nd rectangle: 6

Rectangle with length= 5.0 and width= 3.0 has the lesser area!

Process finished with exit code 0

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

class Time:

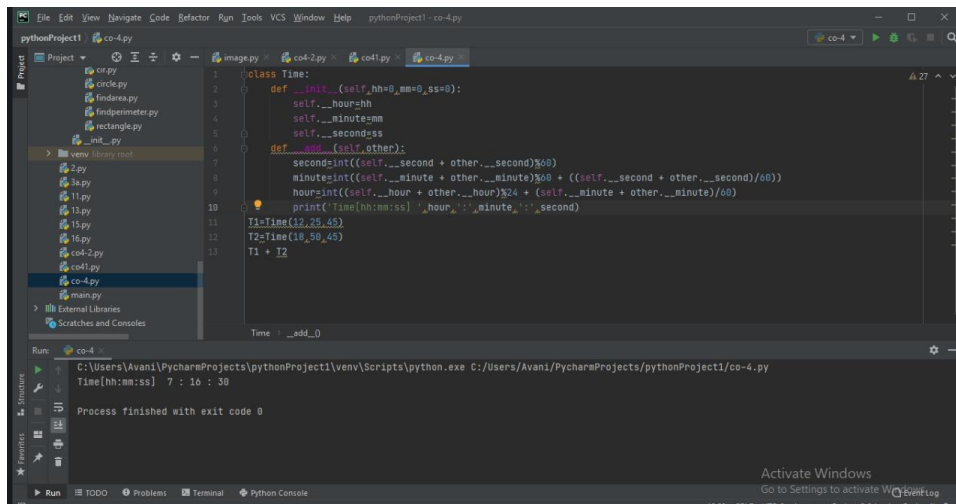
```
def __init__(self,hh=0,mm=0,ss=0):
    self.__hour=hh
    self.__minute=mm
    self.__second=ss
def __add__(self,other):
    second=int((self.__second + other.__second)%60)
    minute=int((self.__minute + other.__minute)%60 + ((self.__second + other.__second)/60))
    hour=int((self.__hour + other.__hour)%24 + (self.__minute + other.__minute)/60)
    print('Time[hh:mm:ss] ',hour,':',minute,':',second)
```

T1=Time(2,25,45)

T2=Time(18,50,45)

T1 + T2

Output

The screenshot shows a Python IDE with a project named 'pythonProject1'. The file 'co-4.py' is open, displaying the 'Time' class implementation. The class has three private attributes: 'hour', 'minute', and 'second'. It includes an 'init' method to set these attributes and an 'add' method to calculate the sum of two time objects. The 'add' method handles carry-over from seconds to minutes and from minutes to hours. The code creates two Time objects, T1 (2:25:45) and T2 (18:50:45), and prints their sum. The output in the console shows the sum as 21:16:30. The IDE interface includes a file explorer on the left, a terminal at the bottom, and a status bar at the bottom right.

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(423.50,302,'An idealist View of Life','Dr.S. Radhakrishnan','Andesite Press')
```

```
P1.show()
```

## Output