

# ASSIGNMENT

## ON

### Python programs

**Submitted to**

**Shelly Shiju George**

**MCA Department**

**Amal Jyothi College of Engineering**

**Submitted by**

**Harithakrishnan**

**1<sup>st</sup> year MCA A**

**Rollno: 40**

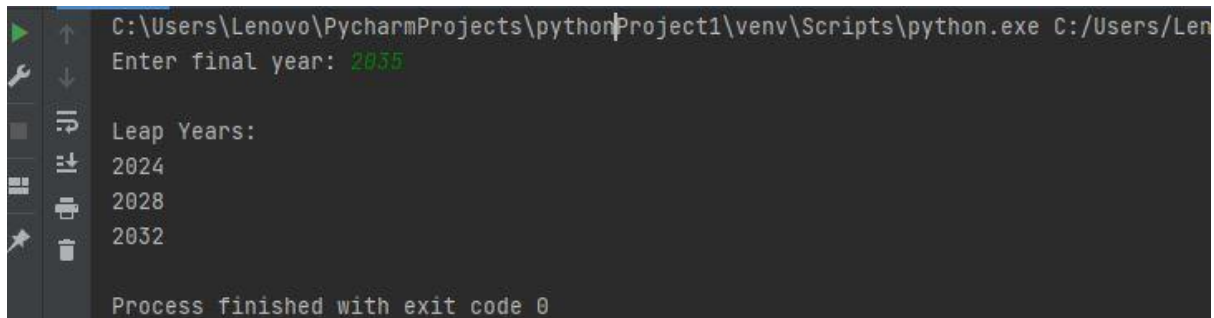
**Amal Jyothi college of Engineering**

**Submitted on 17-02-2021**

**Q1.Display future leap years from current year to a final year entered by user.**

```
import datetime
y=datetime.datetime.now()
y=int(y.year)
lp=int(input("Enter final year: "))
print("\nLeap Years:")
for i in range(y,lp+1):
    if(i%4==0):
        print(i)
```

**output**



```
C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/Lenovo/PycharmProjects/pythonProject1/main.py
Enter final year: 2035

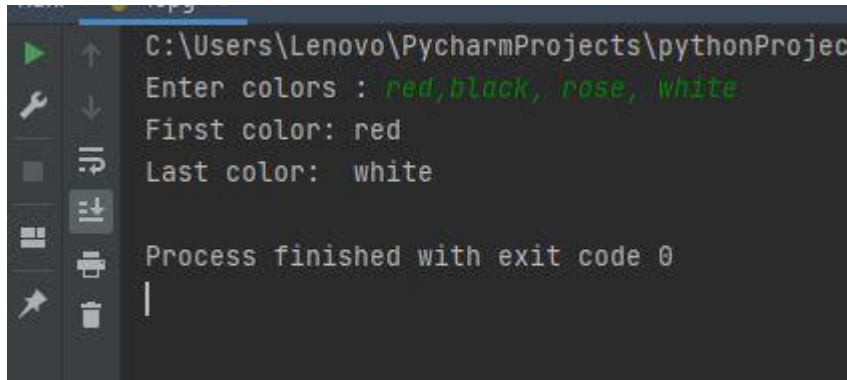
Leap Years:
2024
2028
2032

Process finished with exit code 0
```

**Q2.Create a list of colors from comma-separated color names entered by user. Display first and last colors.**

```
colors=(input('Enter colors : ')).split(',')
print('First color:',colors[0])
print('Last color:',colors[len(colors)-1])
```

Output:



```
C:\Users\Lenovo\PycharmProjects\pythonProject
Enter colors : red,black,rose,white
First color: red
Last color: white
Process finished with exit code 0
```

**Q3.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)))
```

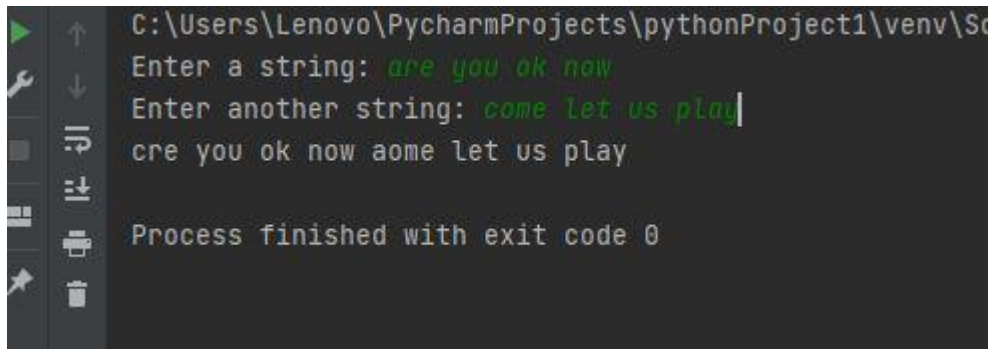
Output:



```
C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users
Enter colors separated by commas: red,green,yellow
Enter colors separated by commas: blue,yellow,black
Colors in color-list1 not contained in color-list2 are: ['green', 'red']
Process finished with exit code 0
```

**Q4. 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)
```



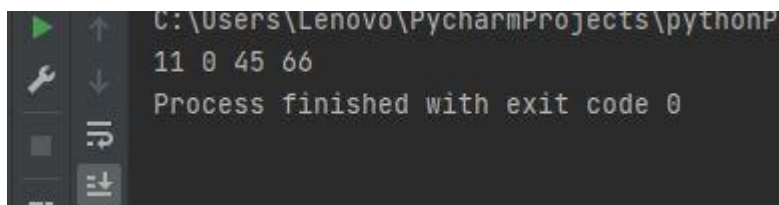
```
C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Sc
Enter a string: are you ok now
Enter another string: come let us play
cre you ok now aome let us play

Process finished with exit code 0
```

#### Q5.Generate positive list of numbers from a given list of integers

```
list1 = [11, -21, 0, 45, 66, -93]
for num in list1:
    if num >= 0:
        print(num, end=" ")
```

#### Output:



```
C:\Users\Lenovo\PycharmProjects\pythonP
11 0 45 66
Process finished with exit code 0
```

**Q6.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)**

Graphics

```
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')
```

```
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)
```

```
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)
```

```
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

```
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')
```

```
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:

```
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
```

**Q7. 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 comp(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(8,2)
r2=Rectangle(5,7)
r1.comp(r2)

```

### Output:



```

C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users
Rectangle with length = 5 and breadth = 7 has the greater area

Process finished with exit code 0

```

**Q8 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

```

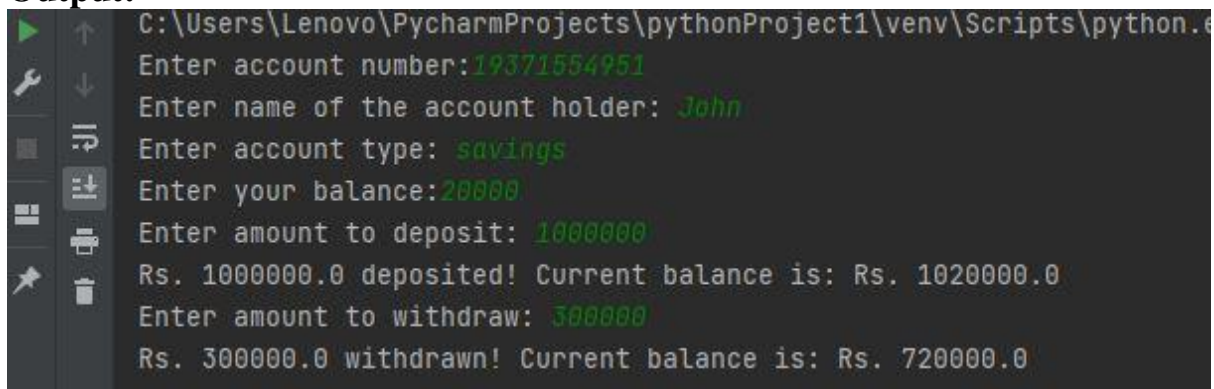


```

    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: ')))

```

### Output:



```

C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe
Enter account number:19371554951
Enter name of the account holder: John
Enter account type: savings
Enter your balance:20000
Enter amount to deposit: 1000000
Rs. 1000000.0 deposited! Current balance is: Rs. 1020000.0
Enter amount to withdraw: 300000
Rs. 300000.0 withdrawn! Current balance is: Rs. 720000.0

```

**Q9. 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

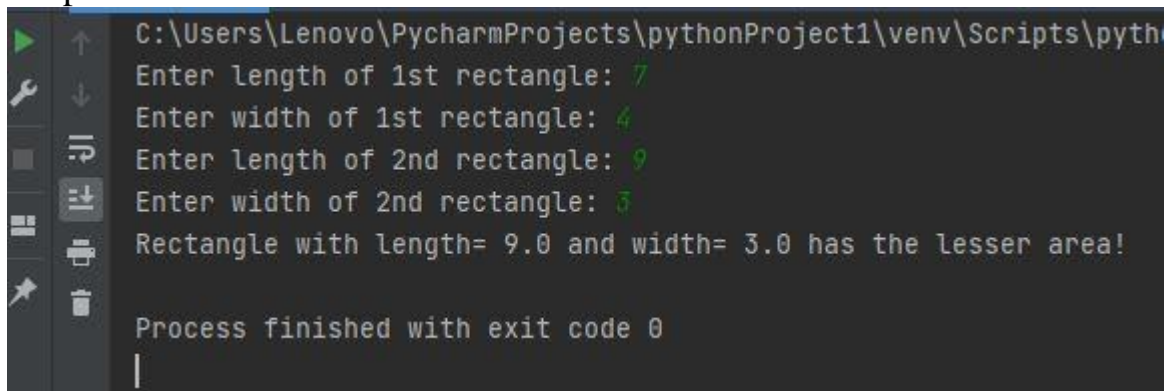
```

```

        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:



```

C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\pyth
Enter length of 1st rectangle: 7
Enter width of 1st rectangle: 4
Enter length of 2nd rectangle: 9
Enter width of 2nd rectangle: 3
Rectangle with length= 9.0 and width= 3.0 has the lesser area!

Process finished with exit code 0

```

**Q10.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)

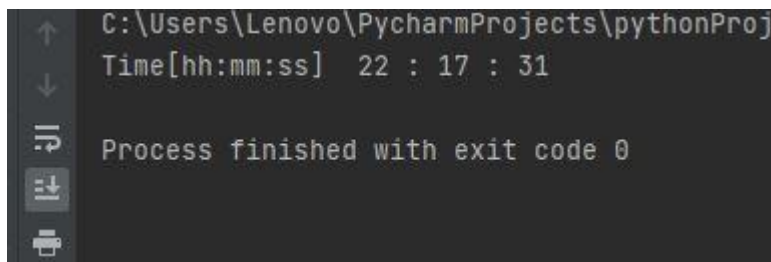
```

```

        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(13,24,45)
T2=Time(8,52,46)
T1 + T2

```

## Output



```

C:\Users\Lenovo\PycharmProjects\pythonProj
Time[hh:mm:ss] 22 : 17 : 31
Process finished with exit code 0

```

**Q11. 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.**

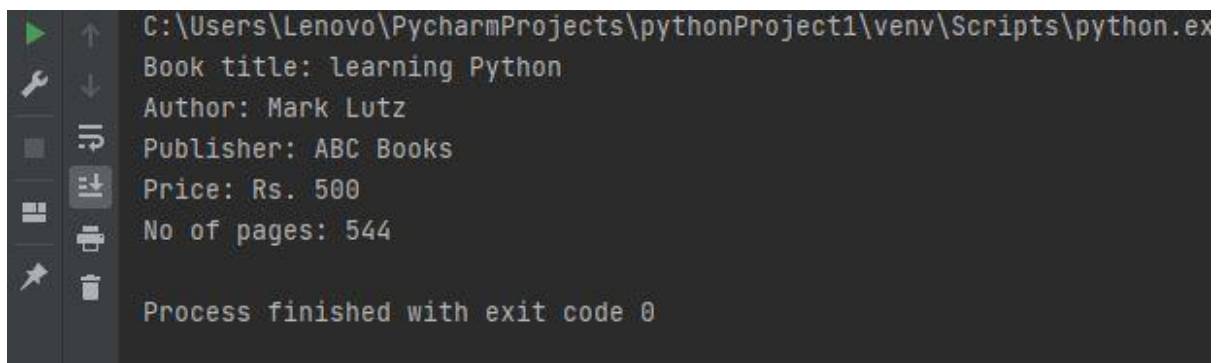
```

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(500,544,'learning Python','Mark Lutz','ABC Books')
P1.show()
```

Output:

A screenshot of a Python IDE terminal window. The terminal shows the output of the code: 'Book title: learning Python', 'Author: Mark Lutz', 'Publisher: ABC Books', 'Price: Rs. 500', and 'No of pages: 544'. The terminal also shows the file path 'C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe' and the message 'Process finished with exit code 0'. The terminal has a dark background and a light-colored text. On the left side of the terminal, there is a vertical toolbar with various icons for running, debugging, and other IDE functions.

```
C:\Users\Lenovo\PycharmProjects\pythonProject1\venv\Scripts\python.exe
Book title: learning Python
Author: Mark Lutz
Publisher: ABC Books
Price: Rs. 500
No of pages: 544
Process finished with exit code 0
```