# 20MCA131 PROGRAMMING LAB

*Lab Report Submitted By*

**MERLIN MONCY**

**Reg. No.:AJC21MCA-2079**

*In Partial fulfilment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 Year)**
**(MCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**


**AMAL JYOTHI COLLEGE OF ENGINEERING**
**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovapally, Kanjirappally, Kottayam, Kerala – 686518]

**2021-2022**

## DEPARTMENT OF COMPUTER APPLICATIONS

## AMAL JYOTHI COLLEGE OF ENGINEERING
## KANJIRAPPALLY

# CERTIFICATE

This is to certify that the lab report, "**20MCA131 PROGRAMMING LAB"** is the bonafide work of **MERLIN MONCY (Reg. No: AJC21MCA-2079)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22

Ms. Rini Kurian

Staff In-Charge

# CONTENT

| | | | |
|---|---|---|---|
| 18 | Find GCD of 2 numbers. | 26-11-2021 | 18 |
| 19 | From a list of integers, create a list removing even numbers. | 26-11-2021 | 19 |
| 20 | Program to find the factorial of a number | 10-12-2021 | 20 |
| 21 | Generate Fibonacci series of N terms | 10-12-2021 | 21 |
| 22 | Find the sum of all items in a list. | 10-12-2021 | 22 |
| 23 | Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square. | 10-12-2021 | 23 |
| 24 | Display the given pyramid with step number accepted from user. | 17-12-2021 | 24 |
| 25 | Count the number of characters (character frequency) in a string. | 17-12-2021 | 25 |
| 26 | Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’ | 17-12-2021 | 26 |
| 27 | Accept a list of words and return length of longest word. | 17-12-2021 | 27 |
| 28 | Construct pattern using nested loop | 17-12-2021 | 28 |
| 29 | Generate all factors of a number. | 17-12-2021 | 29 |
| 30 | Write lambda functions to find area of square, rectangle and triangle. | 7-01-2022 | 30 |
| 31 | Create a package graphics with modules rectangle, circle and sub-package 3D-<br>graphics with modules cuboid and sphere. | 7-01-2022 | 31 |
| 32 | Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area. | 7-01-2022 | 33 |
| 33 | 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. | 7-01-2022 | 35 |
| 34 | Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles. | 7-01-2022 | 37 |
| 35 | Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time. | 7-01-2022 | 39 |
| 36 | Create a class for Book Publisher and performance inheritance. | 7-01-2022 | 41 |
| 37 | Write a Python program to read a file line by line and store it into a list. | 28-01-2022 | 43 |

## Program no:1

## Aim:

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

## Source Code:

```
c= int(input("enter the current year"))
f= int(input("enter the final year"))
print("leap years are :")
for i in range (c,f) :
    if(i%4==0) and (i%100!=0) or (i%400==0) :
        print(i)
```

## Output:

## Program no:2

## Aim:

List comprehensions:
a. Generate positive list of numbers from a given list of integers
b. Square of N numbers
c. Form a list of vowels selected from a given word
d. List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

## Source Code:

```
list1 = [1, 9, 10, 11, -56, 12, 0, 78, -77, 789, -34, 67]
for i in list1:
      if i <= 0:
          list1.remove(i)
print(list1)
list2=[1,2,90,87,100,102,6,1,4]
for i in list1:
   print (i*i)
word=input("enter the word")
j=[ord(x) for x in word]
print(j)
V=['a','e','i','o','u']
s=[i for i in word if i in V]
print(s)
```

## Output:

## Program no:3

## Aim:

Count the occurrences of each word in a line of text.

## Source Code:

```
str1=str(input("enter the string"))
str2=str(input("enter the word "))
x=str1.split()
print(x)
for i in  x :
    if str2 == i :
     c=x.count(i)
print(c)
```

## Output:

## Program no:4

## Aim:

Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

## Source Code:

```
n= int(input("enter the numbers of values"))
a=[]
for i in range(0,n):
    c=int(input("enter the value"))
    if c >100:
        a.append("over")
    else:
        a.append(c)
print(a)
```

## Output:

## Program no:5

## Aim:

Store a list of first names. Count the occurrences of ‘a’ within the list

## Source Code:

```
list1 =['dency','akshara','shalvin','mridhula','nigi']
c=0
for i in list1:
    if "a" in i:
       c=c+1
print(c)
```

## Output:

## Program no:6

## Aim:

Enter 2 lists of integers. Check:
a. Whether list are of same length
b. whether list sums to same value
c. whether any value occur in both

## Source Code:

```
li1=[2,3,4,5,6,7]
li2=[5,6,7,8,10,12]
if len(li1)==len(li2):
    print("a-Length are same\n")
else:
    print("a-lenth are not same")
if sum(li1) == sum(li2):
    print("b-sum are equal")
else:
    print("b-sum are not equal")
j=[x for x in li1 if x in li2]
if j != 0:
    print("c-same elements are",str(j))
else:
    print("c-no elements found")
```

## Output:

## Program no:7

## Aim:

Get a string from an input string where all occurrences of first character replaced with '$', except first character. [eg: onion -> oni$n]

## Source Code:

```
a=input("Enter a string")
print(a[0]+a[1:].replace(a[0],"$"))
```

## Output:

## Program no:8

## Aim:

Create a string from given string where first and last characters exchanged. [eg: python -> nythop]

## Source Code:

```
str1=str(input("enter the word"))
list1=list(str1)
print(list1)
temp=list1[0]
list1[0]=list1[-1]
list1[-1]=temp
print(list1)
```

## Output:

## Program no:9

## Aim:

Accept the radius from user and find area of circle.

## Source Code:

```
from math import pi
r=int(input("enter the radius:"))
print("area of circle=",pi*r*r)
```

## Output:

## Program no:10

## Aim:

Find biggest of 3 numbers entered.

## Source Code:

```
a= int(input("enter the fist number"))
b= int(input("enter the second number"))
c=  int(input("enter the third number"))
if (a>=b) and (b>=c):
   print(a,"is greater")
elif (b>=a ) and (b>=c) :
   print(b,"is greater")
else :
   print(c,"is greater")
```

## Output:

## Program no:11

## Aim:

Accept a file name from user and print extension of that.

## Source Code:

```
 n=input("Enter a filename with extension:")
x=n.split(".")
print("Extension of file name is:",x[-1])
```

## Output:

## Program no:12

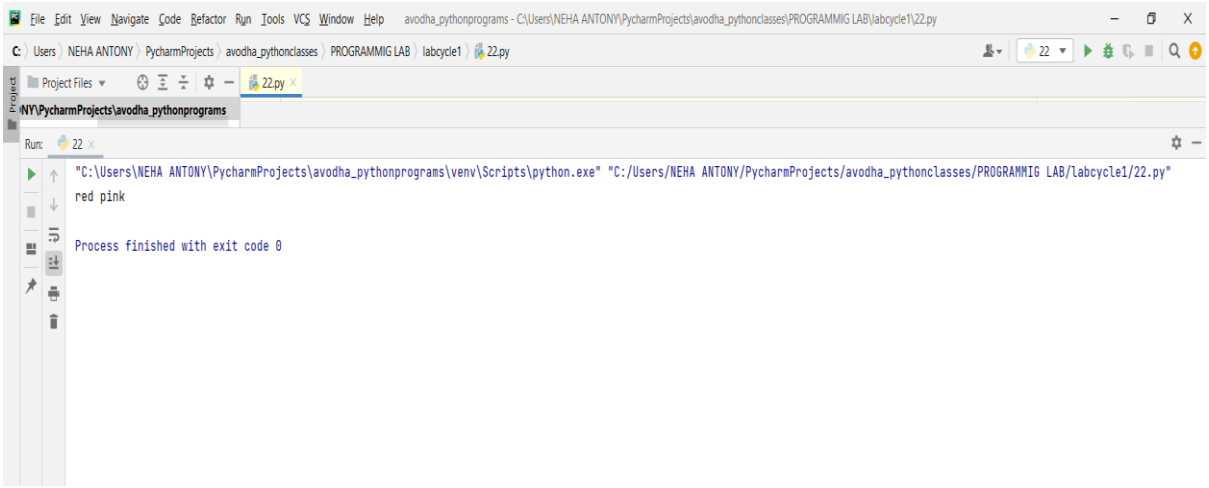
## Aim:

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

## Source Code:

```
colorlist1=["red","blue","green","yellow","pink"]
print(colorlist1[0],colorlist1[-1])
```

## Output:



## Program no:13

## Aim:

Accept an integer n and compute n+nn+nnn.

## Source Code:

```
n = int(input("enter the number"))
print(n+n*n+n*n*n)
```

## Output:

## Program no:14

## Aim:

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

## Source Code:

```
colorlist1=["red","blue","green"]
colorlist2=["red","blue","pink"]
c1=set(colorlist1)
c2=set(colorlist2)
x=c1.difference(c2)
colorl=list(x)
print(colorl)
```

## Output:

## Program no:15

## Aim:

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

## Source Code:

```
a=str(input("enter the str1"))
b=str(input("enter the str2"))
print(a.replace(a[0],b[0])+' '+b.replace(b[0],a[0]))
```

## Output:

## Program no:16

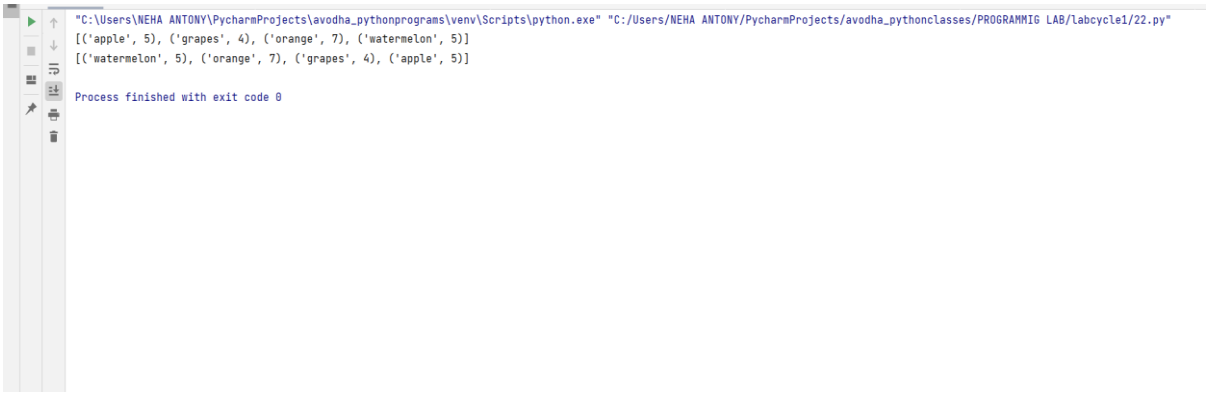
## Aim:

Sort dictionary in ascending and descending order.

## Source Code:

```
fruits={"apple":5,"orange":7,"watermelon":5,"grapes":4}
l=list(fruits.items())
l.sort()
print(l)
l=list(fruits.items())
l.sort(reverse=True)
print(l)
```

## Output:



## Program no:17

## Aim:

Merge two dictionaries.

## Source Code:

```
dict1={1 :"apple",2:"orange",3 : "banana"}
dict2={4 :"plum",5 :"cherry"}
dict1.update(dict2)
print(dict1)
```

## Output:

## Program no: 18

## Aim:

Find gcd of 2 numbers.

## Source Code:

```
n1 = int(input('n1='))
n2= int(input("n2="))
if n1 < n2 :
   small = n1
else :
   small = n2
for i in range (1,small+1):
        if ((n1%i == 0) and (n2%i == 0) ):
            hcf = i

print ("hcf",hcf)
```

## Output:

## Program no:19

## Aim:

From a list of integers, create a list removing even numbers.

## Source Code:

```
list1=[1,2,3,4,5,6,7,8,9,10]
for i in  list1:
      if i % 2==0 :
           list1.remove(i)
print(list1)
```

## Output:

## Program no:20

## Aim:

Program to find the factorial of a number

## Source Code:

```
n=int(input("Enter the number"))
fact=1
for i in range(1,n+1):
    fact=fact*i
print(n,"!=",fact)
```

## Output:

## Program no:21

## Aim:

Generate Fibonacci series of N terms

## Source Code:

```
n=int(input("Enter the number"))
fib=0
print("Fibonacci SERIES:")
for i in range(0,n+1):
    fib=fib+i
    print(fib)
```

## Output:

## Program no:22

## Aim:

Find the sum of all items in a list.

## Source Code:

```
a=[32,322,234,46,7,6]
print(sum(a))
```

## Output:

## Program no:23

## Aim:

Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

## Source Code:

```
st=int(input("Enter the intial range"))
if(st<1000):
    print("enter a 4 dig num")
    st = int(input("Enter the intial range"))
end=int(input("Enter the End range"))
if(end<st):
    print("Enter a value greater than inital range")
    end = int(input("Enter the End range"))
print("Perfect squares and even numbers in the range"+str(st)+"-"+str(end)+":")
for i in range(st,end):
    if i%2==0 and i**(1/2)==int(i**(1/2)):
        print(i)
```

## Output:

## Program no:24

## Aim:

Display the given pyramid with step number accepted from user.
Eg: N=4
1
2 4
3 6 9
4 8 12 16

## Source Code:

```
n=int(input("enter the number"))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(i*j,end=" ")
    print("\n")
```

## Output:

## Program no:25

## Aim:

Count the number of characters (character frequency) in a string.

## Source Code:

```
n=input("Enter the string")
a={}
c=0
for i in n:
    for j in n:
        if i == j:
            c=c+1
    a.update({i:c})
    c=0
print(a.items())
```

## Output:

## Program no:26

## Aim:

Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.

## Source Code:

```
n=input("Enter a string")
if n[len(n)-3:]!="ing" :
   print(n+"ing")
else:
   print(n+"ly")
```

## Output:

## Program no:27

## Aim:

Accept a list of words and return length of longest word.

## Source Code:

```
n=int(input("Enter the no of elements"))
a=[]
for x in range(0,n):
   a.append(input("Enter the word "))
c=0
for i in a:
   if  len(i)>c:
      c=len(i)
      largest=i
print(largest)
```

## Output:

## Program no:28

## Aim:

Construct following pattern using nested loop

```
*
* *
* * *
* * * *
* * *
* *
*
```

## Source Code:

```
n=int(input("enter the number"))
for i in range(1,n+1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
for i in range(n+1,1,-1):
    for j in range(i-1,1,-1):
        print("*",end=" ")
    print("\n")
```

## Output:

## Program no:29

## Aim:

Generate all factors of a number.

## Source Code:

```
n=int(input("Enter the number"))
c=[]
for i in range(1,n+1):
    for j in range(1,i+1):
        if i*j==n:
            c.append(i)
            c.append(j)
print("factors of "+str(n)+" :")
for i in c:
    print(i)
```

## Output:

## Program no:30

## Aim:

Write lambda functions to find area of square, rectangle and triangle.

## Source Code:

```
print("area of rectangle")
l=int(input("length"))
b=int(input("breadth"))
c=lambda x,y: x*y
print("Area of rectangle:"+str(c(l,b)))
print("area of square")
s=int(input("side of square"))
c=lambda x: x*x
print("Area of Square:"+str(c(s)))
print("area of triangle")
l=int(input("base"))
b=int(input("height"))
c=lambda x,y: .5*x*y
print("Area of Square:"+str(c(l,b)))
```

## Output:

## Program no:31

## Aim:

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).

## Source Code:

**graphpack.py**

```
from graphics import circle,rectangle
from graphics.dgraphics import cuboid,sphere
r=int(input("Enter the radius of circle:"))
circle.areac(r)
circle.peric(r)
l=int(input("Enter the length of rectangle:"))
b=int(input("Enter the breadth of rectangle:"))
rectangle.arear(l,b)
rectangle.perir(l,b)
l1=int(input("Enter the length of cuboid:"))
b1=int(input("Enter the breadth of cuboid:"))
h1=int(input("Enter the height of cuboid:"))
cuboid.areacub(l1,b1,h1)
cuboid.pericub(l1,b1,h1)
r1=int(input("Enter the radius of sphere:"))
sphere.areas(r1)
sphere.peris(r1)
```

**graphics**
**circle.py**

```
def areac(r):
    a=3.14*r*r
    print("Area of Circle is:",a)
def peric(r):
    p=2*3.14*r
    print("Perimeter of Circle is:",p)
```

**rectangle.py**

```
def arear(l,b):
    a=l*b
    print("Area of Rectangle is:",a)
def perir(l,b):
    p=2*(l+b)
    print("Area of Rectangle is:",p)
```

**dgraphics**

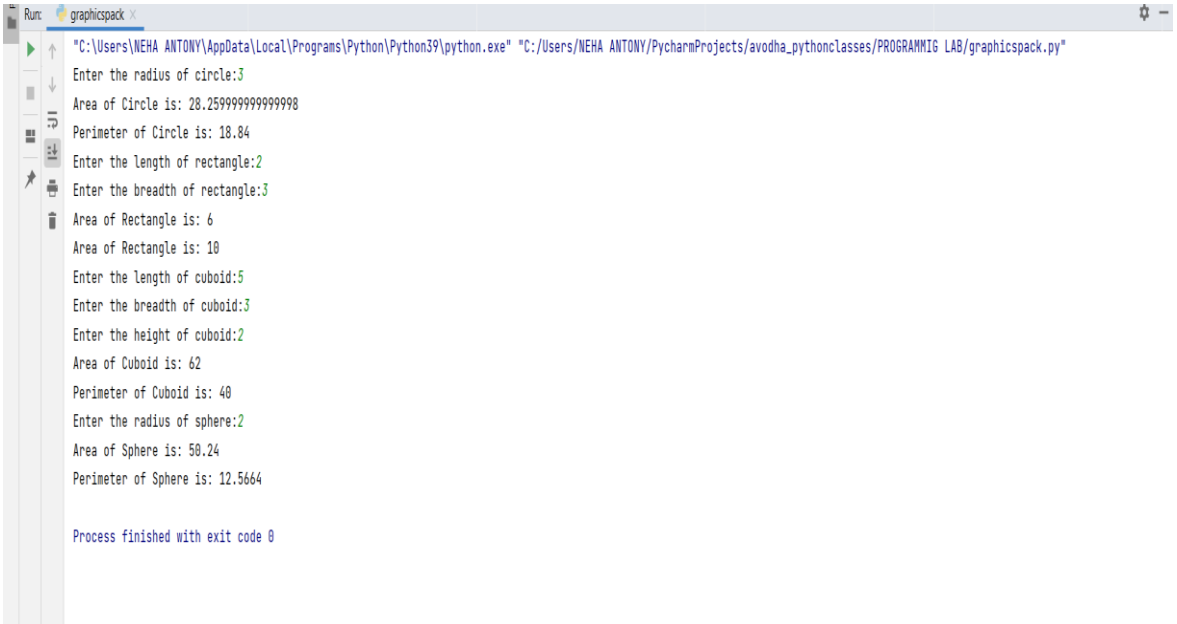
**sphere.py**

```
def areas(r):
    a = 4*3.14*r*r
    print("Area of Sphere is:", a)
def peris(r):
    p = 6.2832*r
    print("Perimeter of Sphere is:", p)
```

**cuboid.py**

```
def areacub(l,b,h):
    a = 2*((l*b) + (b*h) + (h*l))
    print("Area of Cuboid is:", a)
def pericub(l,b,h):
    p = 4*(l+b+h)
    print("Perimeter of Cuboid is:", p)
```

## Output:



## Program no: 32

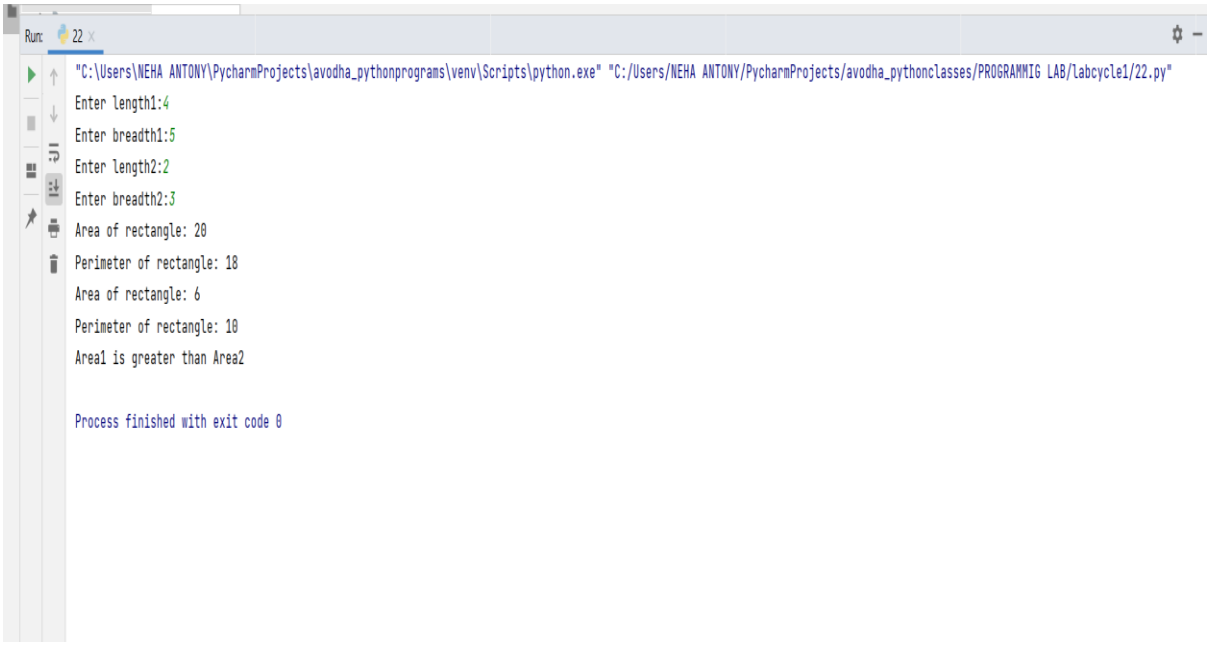
## Aim:

Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

## Source Code:

```
class rect:
    def __init__(self,l,b):
        self.a1=l
        self.a2=b
    def area(self):
        self.m=self.a1*self.a2
    def peri(self):
        self.n=2*(self.a1 + self.a2)
    def disp(self):
        print("Area of rectangle:", self.m)
        print("Perimeter of rectangle:", self.n)
    def compare(self,obj2):
        if self.m == obj2.m:
            print("Areas are equal")
        elif self.m > obj2.m:
            print("Area1 is greater than Area2")
        else:
            print("Area2 is greater than Area1")
l1=int(input("Enter length1:"))
b1=int(input("Enter breadth1:"))
l2=int(input("Enter length2:"))
b2=int(input("Enter breadth2:"))
obj1=rect(l1,b1)
obj2=rect(l2,b2)
obj1.area()
obj1.peri()
obj2.area()
obj2.peri()
obj1.disp()
obj2.disp()
obj1.compare(obj2)
```

## Output:


Run: 22
"C:\Users\NEHA ANTONY\PycharmProjects\avodha_pythonprograms\venv\Scripts\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/labcycle1/22.py"
Enter length1:4
Enter breadth1:5
Enter length2:2
Enter breadth2:3
Area of rectangle: 20
Perimeter of rectangle: 18
Area of rectangle: 6
Perimeter of rectangle: 10
Area1 is greater than Area2
Process finished with exit code 0

## Program no: 33

## Aim:

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.

## Source Code:

```
class bank:
    def __init__(self,a,n,t,b):
        self.ac = a
        self.name = n
        self.type = t
        self.bal = b
    def depo(self,a1):
        self.bal += a1
        print("Balance:",self.bal)
    def widthdraw(self,a2):
        if self.bal<a2:
            print("Invalid")
        else:
            self.bal -= a2
            print("Balance:",self.bal)
    def disp(self):
        print("Acc No:",self.ac)
        print("Name:", self.name)
        print("Acc Type:", self.type)
        print("Acc Balance:", self.bal)
a=int(input("Enter acc no:"))
n=input("Enter name:")
t=input("Enter acc type:")
b=int(input("Enter balance:"))
obj1=bank(a,n,t,b)
obj1.disp()
a1=int(input("Enter the amount to deposite:"))
obj1.depo(a1)
a2=int(input("Enter the amount to widthdraw:"))
obj1.widthdraw(a2)
```

## Output:


"C:\Users\NEHA ANTONY\PycharmProjects\avodha_pythonprograms\venv\Scripts\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/labcycle1/22.py"
Enter acc no:3456
Enter name:riya
Enter acc type:persnol
Enter balance:456789
Acc No: 3456
Name: riya
Acc Type: persnol
Acc Balance: 456789
Enter the amount to deposite:1000
Balance: 457789
Enter the amount to widthdraw:3000
Balance: 454789
Process finished with exit code 0

## Program no: 34

**Aim:** Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

## Source Code:

```
class Rectangle:
 def __init__(self, l, b):
  self._l1 = l
  self._b1 = b
 def area(self):
  area1 = self._l1 * self._b1
  return area1
 def __lt__(self, obj):
   if (self.area() < obj.area()):
     return "The area of Rectangle1 is less than Rectangle2"
   else:
     return "The area of Rectangle2 is less than Rectangle1"

print("RECTANGLE 1")
l = int(input("Enter the length of rectangle1:"))
b = int(input("Enter the breadth of rectangle1:"))
obj1 = Rectangle(l,b)
print("The area is:")
print(obj1.area())
print("RECTANGLE 2")
l=int(input("Enter the length of rectangle2:"))
b=int(input("Enter the breadth of rectangle3:"))
obj2 = Rectangle(l,b)
print("The area is:")
print(obj2.area())
print("Now Comparing The Rectangles")
print(obj1 < obj2)
```

## Output:


"C:\Users\NEHA ANTONY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/pr.py"
RECTANGLE 1
Enter the length of rectangle1:2
Enter the breadth of rectangle1:2
The area is:
4
RECTANGLE 2
Enter the length of rectangle2:3
Enter the breadth of rectangle3:3
The area is:
9
Now Comparing The Rectangles
The area of Rectangle1 is less than Rectangle2
Process finished with exit code 0

## Program no: 35

## Aim:

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

## Source Code:

```
class Time:
  def __init__(self, h, m, s):
   self._h1 = h
   self._m1 = m
   self._s1 = s
  def __add__(self, x):
   sum1 = self._h1 + x._h1
   sum2 = self._m1 + x._m1
   sum3 = self._s1 + x._s1
   if sum3 >= 60:
     sum3 = sum3 - 60
     sum2 =sum2 + 1
   if sum2 >= 60:
    sum2 = sum2 - 60
    sum1 = sum1 + 1
    print(sum1, ":", sum2,":", sum3);

print("TIME 1")
h1 = int(input("Enter the hour in time1:"))
m1 = int(input("Enter the minute in time1:"))
s1 = int(input("Enter the second in time1:"))
obj1 = Time(h1, m1, s1)
print("TIME 2")
h2 = int(input("Enter the hour in time2:"))
m2 = int(input("Enter the minute in time2:"))
s2 = int(input("Enter the second in time2:"))
obj2 = Time(h2, m2, s2)
print("The sum of both time are:")
obj1 + obj2
```

## Output:


Run:
pr
"C:\Users\NEHA ANTONY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/pr.py"
TIME 1
Enter the hour in time1:4
Enter the minute in time1:56
Enter the second in time1:34
TIME 2
Enter the hour in time2:7
Enter the minute in time2:45
Enter the second in time2:10
The sum of both time are:
12 : 41 : 44
Process finished with exit code 0

## Program no: 36

## Aim:

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.

## Source Code:

```
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(700,300,'Programming with Python','GV Rossum','ABC Books')
P1.show()
```

## Output:


Run:
pr
"C:\Users\NEHA ANTONY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG L
Book title: Programming with Python
Author: GV Rossum
Publisher: ABC Books
Price: Rs. 700
No of pages: 300
Process finished with exit code 0

## Program no: 37

## Aim:

Write a Python program to read a file line by line and store it into a list.

## Source Code:

**demo.txt**

```
Python
Interpreted high-level language.
Python is object oriented programming language
```

**line.py**

```
def fread(fname):
    with open(fname) as f:
        c = f.readlines()
    print(c)
fread("demo")
```

## Output:

## Program no: 38

## Aim:

Python program to copy odd lines of one file to other

## Source Code:

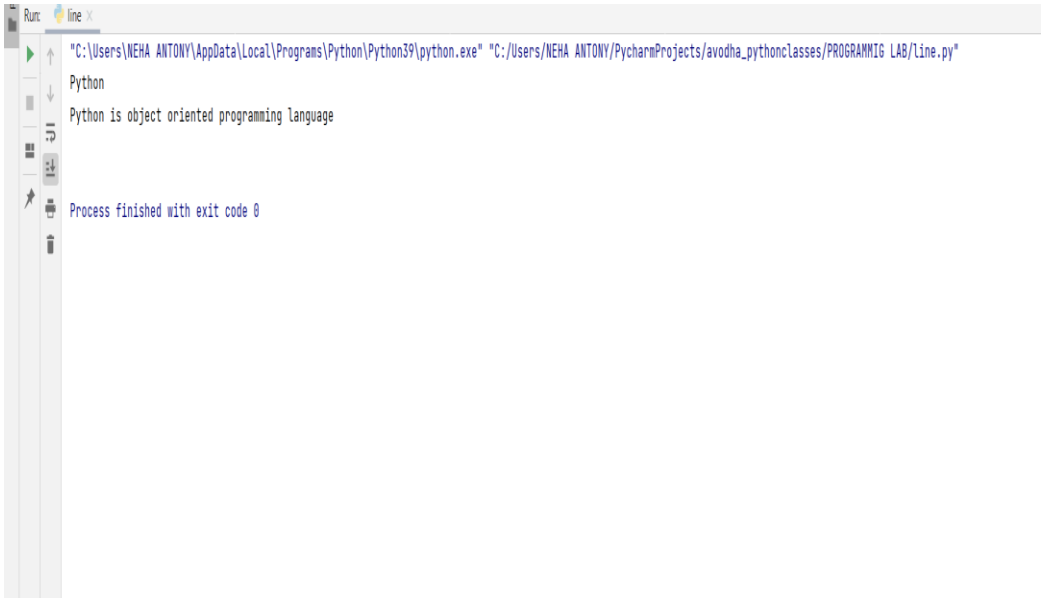
**demo.txt**

```
Python
Interpreted high-level language.
Python is object oriented programming language
```

**line.py**

```
a = open("demo", "r")
b = open("t", "w")
c = a.readlines()
d = len(c)
for i in range(0, d):
    if i % 2 == 0:
        b.write(c[i])
    else:
        pass
b.close()
b = open("t", "r")
e = b.read()
print(e)
a.close()
b.close()
-
```

## Output:

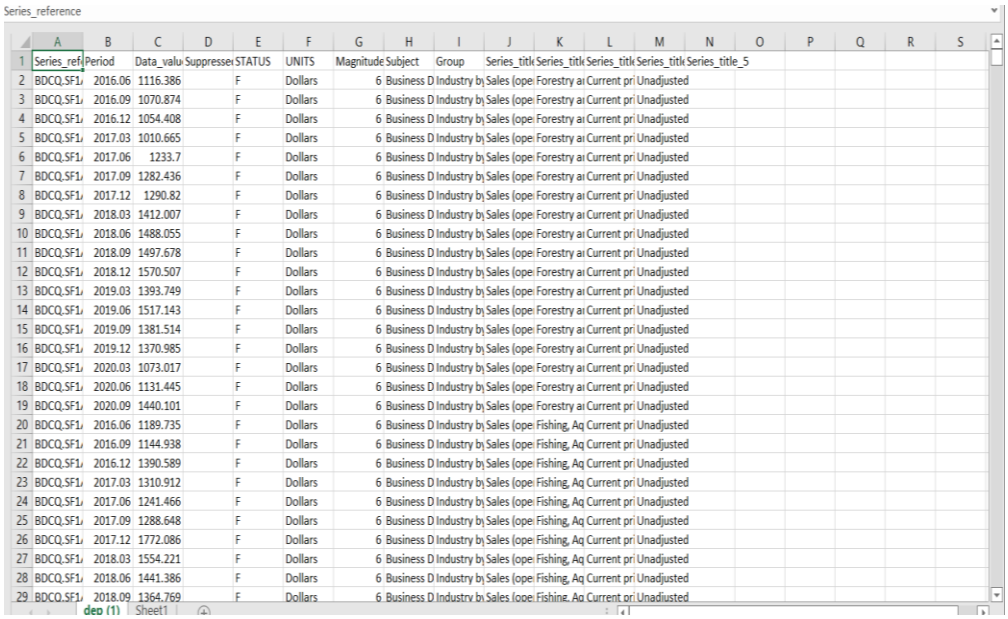


# Program no: 39

**Aim:** Write a Python program to read each row from a given csv file and print a list of strings.

## Source Code:

## CSV file



Series_reference

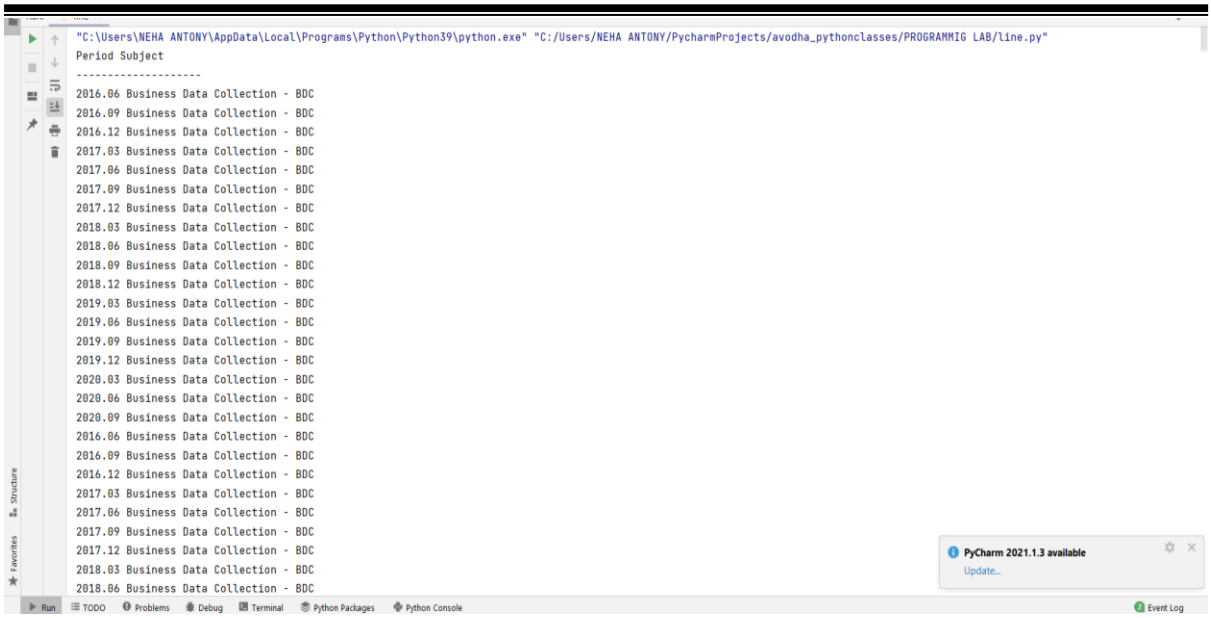
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Series_ref | Period | Data_valu | Suppresse | STATUS | UNITS | Magnitude | Subject | Group | Series_titl | Series_titl | Series_titl | Series_titl | Series_title_5 | | | | | |
| 2 | BDCQ.SF1 | 2016.06 | 1116.386 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 3 | BDCQ.SF1 | 2016.09 | 1070.874 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 4 | BDCQ.SF1 | 2016.12 | 1054.408 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 5 | BDCQ.SF1 | 2017.03 | 1010.665 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 6 | BDCQ.SF1 | 2017.06 | 1233.7 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 7 | BDCQ.SF1 | 2017.09 | 1282.436 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 8 | BDCQ.SF1 | 2017.12 | 1290.82 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 9 | BDCQ.SF1 | 2018.03 | 1412.007 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 10 | BDCQ.SF1 | 2018.06 | 1488.055 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 11 | BDCQ.SF1 | 2018.09 | 1497.678 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 12 | BDCQ.SF1 | 2018.12 | 1570.507 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 13 | BDCQ.SF1 | 2019.03 | 1393.749 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 14 | BDCQ.SF1 | 2019.06 | 1517.143 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 15 | BDCQ.SF1 | 2019.09 | 1381.514 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 16 | BDCQ.SF1 | 2019.12 | 1370.985 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 17 | BDCQ.SF1 | 2020.03 | 1073.017 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 18 | BDCQ.SF1 | 2020.06 | 1131.445 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 19 | BDCQ.SF1 | 2020.09 | 1440.101 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 20 | BDCQ.SF1 | 2016.06 | 1189.735 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 21 | BDCQ.SF1 | 2016.09 | 1144.938 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 22 | BDCQ.SF1 | 2016.12 | 1390.589 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 23 | BDCQ.SF1 | 2017.03 | 1310.912 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 24 | BDCQ.SF1 | 2017.06 | 1241.466 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 25 | BDCQ.SF1 | 2017.09 | 1288.648 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 26 | BDCQ.SF1 | 2017.12 | 1772.086 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 27 | BDCQ.SF1 | 2018.03 | 1554.221 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 28 | BDCQ.SF1 | 2018.06 | 1441.386 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 29 | BDCQ.SF1 | 2018.09 | 1364.769 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |

dep (1) Sheet1

**line.py**

```
import csv
with open("csv", newline='') as csvfile:
    d = csv.reader(csvfile, delimiter=' ', quotechar='|')
    for i in d:
        print(', '.join(i))
```

## Output:


"C:\Users\NEHA ANTONY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/line.py"
Period Subject
2016.06 Business Data Collection - BDC
2016.09 Business Data Collection - BDC
2016.12 Business Data Collection - BDC
2017.03 Business Data Collection - BDC
2017.06 Business Data Collection - BDC
2017.09 Business Data Collection - BDC
2017.12 Business Data Collection - BDC
2018.03 Business Data Collection - BDC
2018.06 Business Data Collection - BDC
2018.09 Business Data Collection - BDC
2018.12 Business Data Collection - BDC
2019.03 Business Data Collection - BDC
2019.06 Business Data Collection - BDC
2019.09 Business Data Collection - BDC
2019.12 Business Data Collection - BDC
2020.03 Business Data Collection - BDC
2020.06 Business Data Collection - BDC
2020.09 Business Data Collection - BDC
2016.06 Business Data Collection - BDC
2016.09 Business Data Collection - BDC
2016.12 Business Data Collection - BDC
2017.03 Business Data Collection - BDC
2017.06 Business Data Collection - BDC
2017.09 Business Data Collection - BDC
2017.12 Business Data Collection - BDC
2018.03 Business Data Collection - BDC
2018.06 Business Data Collection - BDC
PyCharm 2021.1.3 available
Update...
Run
TODO
Problems
Debug
Terminal
Python Packages
Python Console
Event Log
Structure
Favorites

## Program no: 40

## Aim:

Write a Python program to read specific columns of a given CSV file and print the content of the columns.

## Source Code:

## CSV file

Series_reference

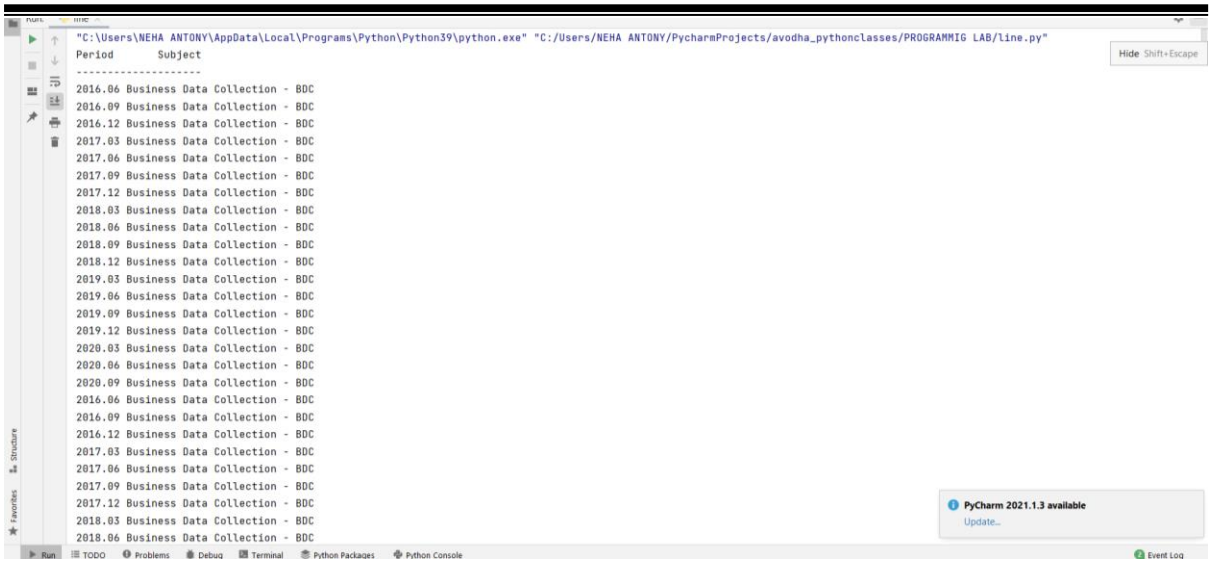
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Series_ref | Period | Data_valu | Suppresse | STATUS | UNITS | Magnitude | Subject | Group | Series_titl | Series_titl | Series_titl | Series_titl | Series_title_5 | | | | | |
| 2 | BDCQ.SF1 | 2016.06 | 1116.386 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 3 | BDCQ.SF1 | 2016.09 | 1070.874 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 4 | BDCQ.SF1 | 2016.12 | 1054.408 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 5 | BDCQ.SF1 | 2017.03 | 1010.665 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 6 | BDCQ.SF1 | 2017.06 | 1233.7 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 7 | BDCQ.SF1 | 2017.09 | 1282.436 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 8 | BDCQ.SF1 | 2017.12 | 1290.82 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 9 | BDCQ.SF1 | 2018.03 | 1412.007 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 10 | BDCQ.SF1 | 2018.06 | 1488.055 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 11 | BDCQ.SF1 | 2018.09 | 1497.678 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 12 | BDCQ.SF1 | 2018.12 | 1570.507 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 13 | BDCQ.SF1 | 2019.03 | 1393.749 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 14 | BDCQ.SF1 | 2019.06 | 1517.143 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 15 | BDCQ.SF1 | 2019.09 | 1381.514 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 16 | BDCQ.SF1 | 2019.12 | 1370.985 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 17 | BDCQ.SF1 | 2020.03 | 1073.017 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 18 | BDCQ.SF1 | 2020.06 | 1131.445 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 19 | BDCQ.SF1 | 2020.09 | 1440.101 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 20 | BDCQ.SF1 | 2016.06 | 1189.735 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 21 | BDCQ.SF1 | 2016.09 | 1144.938 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 22 | BDCQ.SF1 | 2016.12 | 1390.589 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 23 | BDCQ.SF1 | 2017.03 | 1310.912 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 24 | BDCQ.SF1 | 2017.06 | 1241.466 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 25 | BDCQ.SF1 | 2017.09 | 1288.648 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 26 | BDCQ.SF1 | 2017.12 | 1772.086 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 27 | BDCQ.SF1 | 2018.03 | 1554.221 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 28 | BDCQ.SF1 | 2018.06 | 1441.386 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 29 | BDCQ.SF1 | 2018.09 | 1364.769 | | F | Dollars | 6 | Business D | Industry b | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |

dep (1) Sheet1

**line.py**

```
import csv
with open("csv", newline='') as csvfile:
    d = csv.DictReader(csvfile)
    print("Period       Subject")
    print("--------------------")
    for i in d:
        print(i['Period'], i['Subject'])
```

## Output:

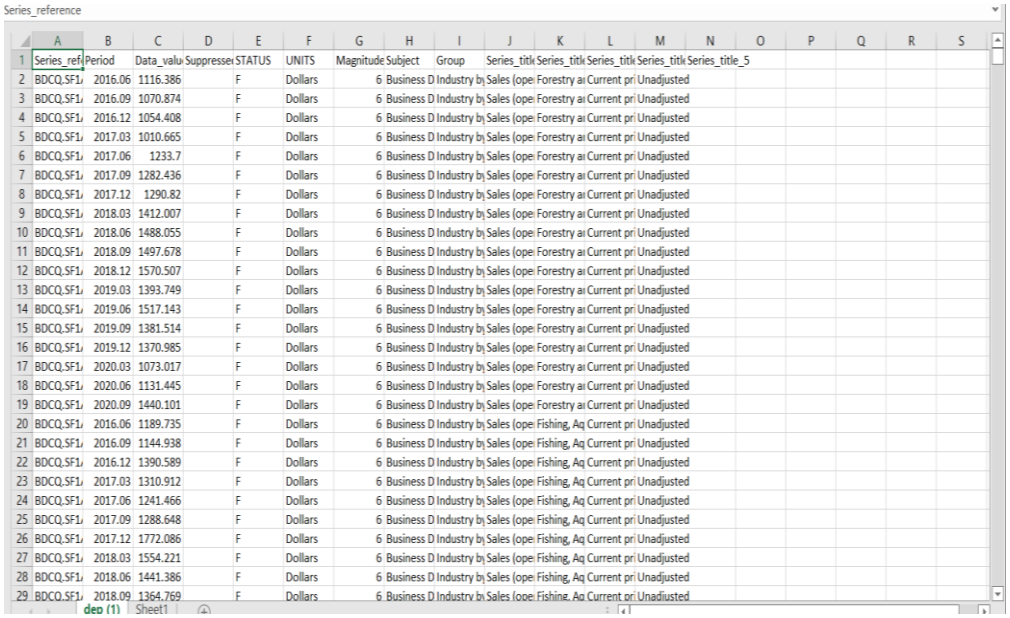

"C:\Users\NEHA ANTONY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/line.py"
Period Subject
2016.06 Business Data Collection - BDC
2016.09 Business Data Collection - BDC
2016.12 Business Data Collection - BDC
2017.03 Business Data Collection - BDC
2017.06 Business Data Collection - BDC
2017.09 Business Data Collection - BDC
2017.12 Business Data Collection - BDC
2018.03 Business Data Collection - BDC
2018.06 Business Data Collection - BDC
2018.09 Business Data Collection - BDC
2018.12 Business Data Collection - BDC
2019.03 Business Data Collection - BDC
2019.06 Business Data Collection - BDC
2019.09 Business Data Collection - BDC
2019.12 Business Data Collection - BDC
2020.03 Business Data Collection - BDC
2020.06 Business Data Collection - BDC
2020.09 Business Data Collection - BDC
2016.06 Business Data Collection - BDC
2016.09 Business Data Collection - BDC
2016.12 Business Data Collection - BDC
2017.03 Business Data Collection - BDC
2017.06 Business Data Collection - BDC
2017.09 Business Data Collection - BDC
2017.12 Business Data Collection - BDC
2018.03 Business Data Collection - BDC
2018.06 Business Data Collection - BDC
Hide Shift+Escape
PyCharm 2021.1.3 available
Update...
Run
TODO
Problems
Debug
Terminal
Python Packages
Python Console
Event Log

## Program no: 41

**Aim:** Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

## Source Code:

## CSV file



Series_reference

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Series_ref | Period | Data_valu | Suppresse | STATUS | UNITS | Magnitude | Subject | Group | Series_titl | Series_titl | Series_titl | Series_titl | Series_title_5 | | | | | |
| 2 | BDCQ.SF1 | 2016.06 | 1116.386 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 3 | BDCQ.SF1 | 2016.09 | 1070.874 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 4 | BDCQ.SF1 | 2016.12 | 1054.408 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 5 | BDCQ.SF1 | 2017.03 | 1010.665 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 6 | BDCQ.SF1 | 2017.06 | 1233.7 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 7 | BDCQ.SF1 | 2017.09 | 1282.436 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 8 | BDCQ.SF1 | 2017.12 | 1290.82 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 9 | BDCQ.SF1 | 2018.03 | 1412.007 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 10 | BDCQ.SF1 | 2018.06 | 1488.055 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 11 | BDCQ.SF1 | 2018.09 | 1497.678 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 12 | BDCQ.SF1 | 2018.12 | 1570.507 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 13 | BDCQ.SF1 | 2019.03 | 1393.749 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 14 | BDCQ.SF1 | 2019.06 | 1517.143 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 15 | BDCQ.SF1 | 2019.09 | 1381.514 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 16 | BDCQ.SF1 | 2019.12 | 1370.985 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 17 | BDCQ.SF1 | 2020.03 | 1073.017 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 18 | BDCQ.SF1 | 2020.06 | 1131.445 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 19 | BDCQ.SF1 | 2020.09 | 1440.101 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Forestry a | Current pri | Unadjusted | | | | | | |
| 20 | BDCQ.SF1 | 2016.06 | 1189.735 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 21 | BDCQ.SF1 | 2016.09 | 1144.938 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 22 | BDCQ.SF1 | 2016.12 | 1390.589 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 23 | BDCQ.SF1 | 2017.03 | 1310.912 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 24 | BDCQ.SF1 | 2017.06 | 1241.466 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 25 | BDCQ.SF1 | 2017.09 | 1288.648 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 26 | BDCQ.SF1 | 2017.12 | 1772.086 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 27 | BDCQ.SF1 | 2018.03 | 1554.221 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 28 | BDCQ.SF1 | 2018.06 | 1441.386 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |
| 29 | BDCQ.SF1 | 2018.09 | 1364.769 | | F | Dollars | 6 | Business D | Industry by | Sales (ope | Fishing, Aq | Current pri | Unadjusted | | | | | | |

dep (1) Sheet1

**line.py**

```
import csv
field_name = ['No', 'Company', 'Car Model']
car = [{'No': 1,'Company':'Ferrari','Car Model':'GH'},
       {'No': 2,'Company':'BMW','Car Model':'X5'},
       {'No': 3,'Company':'Maruti Suzuki','Car Model': 'Swift'},
       {'No': 4,'Company':'Audi', 'Car Model':'RS7'},
       {'No': 5,'Company':'Toyota', 'Car Model':'Fortuner'}]
with open('b.csv', 'w') as csvfile:
   write = csv.DictWriter(csvfile, fieldnames=field_name)
   write.writeheader()
   write.writerows(car)
with open('b.csv', newline='') as csvfile:
   d = csv.reader(csvfile, delimiter='|')
   for r in d:
      print(','.join(r))
```

## Output:


Run:
line
"C:\Users\NEHA ANTONY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/NEHA ANTONY/PycharmProjects/avodha_pythonclasses/PROGRAMMIG LAB/line.py"
No,Company,Car Model
1,Ferrari,GH
2,BMW,X5
3,Maruti Suzuki,Swift
4,Audi,RS7
5,Toyota,Fortuner
Process finished with exit code 0
Structure