FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

PYTHON PROGRAMMING LAB

……..

LABORATORY RECORD

Name: ABHINAV H

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch: 2021 A Roll No: 03

FEDERAL INSTITUTE OF

SCIENCE AND TECHNOLOGY

(FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name :ABHINAV H

Branch : MASTER OF COMPUTER APPLICATIONS

Semester : 1 Roll No: 03

University Exam.Reg. No: FIT21MCA-2003

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done and submitted to Kerala Technological University *in partial fulfillment for the award of the Master Of Computer Applications is a record of the original research work done by* ***ABHINAV H*** *in the* ***PYTHON*** *Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.*

Signature of Staff in Charge — Signature of H.O.D

Name: JOICE T — Name: DEEPA MARY MATHEWS

Date:

Date of University practical examination ……

Signature of Internal Examiner — Signature of External Examiner

CONTENT

10		Accept a file name from user and print extension of that.		
11		Create a list of colors from comma-separated color names entered by user. Display first and last colors.		
12		Accept an integer n and compute n+nn+nnn.		
13		Print out all colors from color-list1 not contained in color-list2.		
14		Create a single string separated with space from two strings by swapping the character at position 1.		
15		Merge two dictionaries.		
16		Find gcd of 2 numbers.		
17		From a list of integers, create a list removing even numbers.		
18		Program to find the factorial of a number.		
19		Generate Fibonacci series of N terms.		
20		Find the sum of all items in a list.		
21		Generate a list of four-digit numbers in a given range with all their digits even and the number is a perfect square.		
22		Display the given pyramid with the step number accepted from the user.		
23		Count the number of characters (character frequency) in a string.		

24		Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.		
25		Accept a list of words and return length of longest word.		
26		Construct following pattern using nested loop.		
27		Generate all factors of a number.		
28		Create a package graphics with modules rectangle, circle and sub-package 3D graphics with module 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.		
29		Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare to rectangle objects by their area.		
30		Create 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.		
31		Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of two rectangles.		
32		Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of two time.		

33		Create a class Publisher(name). Derive class Book from Publisher with attributes title and author. Derive class python from Book with attributes price and number_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overreading.		
34		Write a program to read a file line by line and store it into a list		
35		Write a Python program to read each row from a given csv file and print a list of strings.		

COURSE OUTCOME-1

EXPERIMENT 1

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

CODE:

```
current_year=int(input("Enter the current year:")) f
inal_year=int(input("Enter the final year:"))
 for year in range(current_year,final_year):
            if(year%400==0)or(year%100!=0)and(year%4==0):
                        print(year)
```

OUTPUT:

```
ccf@FISATPC0360:~/abhinav/PYTHON$ python3 leap.py
print leap year between two given year
Enter start year2016
Enter last year2050
list of leap years
2016
2020
2024
2028
2032
2036
2040
2044
2048
ccf@FISATPC0360:~/abhinav/PYTHON$
```

EXPERIMENT 2

AIM: List comprehensions:

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

CODE:

```
      list1=[12,-3,0,4,6]
for num in list1:
        if(num>=0):
                print(num)
```

OUTPUT

```
stud@debian:~/Abhinav/PYTHON$ python3 comprehena.py
12
0
4
6
```

AIM:

b.Print square of N numbers.

CODE:

```
n=int(input("enter the range"))
  for num in range(1,n+1):
        num=num*num
        print(num)
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON$ python3 comprehenb.py
enter the range5
1
4
9
16
25
```

AIM:

c. Form a list of vowels selected from a given word.

CODE:

```
s=input("Enter the word:")
list=[]
for i in s:
        if i in "aeiouAEIOU":
                list.append(i)
print(list)
```

OUTPUT:

```
stud@debian:~$ python3 comberhence.py
Enter the word:abhinav
['a', 'i', 'a']
```

AIM:

d.List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

CODE:

```
list=['F','I','S','A','T']
for i in range(0,5):
        value=ord(list[i])
        print(value)
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON$ python3 comprehend.py
70
73
83
65
84
```

EXPERIMENT 3

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

CODE:

```
list1=[]
list2=[]
x=input("Enter a line of text:")
for i in x.split(" "):
        list1.append(i)
        if i not in list2:
                list2.append(i)
for i in list2:
        print(i,"\t",list1.count(i))
```

OUTPUT:

```
Enter a line of text:how are you
how       1
how       1
are       1
how       1
are       1
you       1


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 4

AIM: Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

CODE:

```
list=[]
n=int(input("Enter the limit:"))
print("Enter integer Numbers")
for i in range(0,n):
        j=int(input())
        if j>100:
                list.append("over")
        else:
                list.append(j)
print(list)
```

OUTPUT:

```
ccf@FISATPC0360: ~/abhinav/PYTHON
File  Edit  View  Search  Terminal  Help
ccf@FISATPC0360:~/abhinav/PYTHON$ gedit greater.py
ccf@FISATPC0360:~/abhinav/PYTHON$ python3 greater.py
Enter the limit:5
Enter integer Numbers
56
45
26
24
12
546
[56, 45, 26, 24, 12, 'over']
ccf@FISATPC0360:~/abhinav/PYTHON$ python3 greater.py
Enter the limit:4
Enter integer Numbers
45
56
23
24
[45, 56, 23, 24]
ccf@FISATPC0360:~/abhinav/PYTHON$
```

EXPERIMENT 5

AIM: Store a list of first names. Count the occurrences of 'a' within the list

CODE:

```
list=['abhi','vyshnav','adrash'] print("Elements in the list are:")
print(list)
count=0
for word in list:
    for i in word:
        if i=='a':
            count+=1
print("count of 'a' is:", count)
```

OUTPUT:

```
Elements in the list are:
['abhi', 'vyshnav', 'adrash']
count of 'a' is: 1
count of 'a' is: 2
count of 'a' is: 3
count of 'a' is: 4
```

EXPERIMENT 6

1) AIM: Enter 2 lists of integers.Check

- **a.** whether list are of same length
- **b.** whetherlist sums of same value
- **c.** whether any value occur in both.

CODE:

```
l1=[1,2,3,4]
l2=[1,3,2]
print("List 1",l1)
print("List 2",l2)
x=len(l1)
```

```
y=len(l2)
if x==y:
 print("List are of same length")
else:
 print("Length of lists are different")
s1=0
s2=0
for i in range(x):
 s1=s1+l1[i]
print("Sum of elements of List1:",s1)
for j in range(y):
 s2=s2+l2[j]
print("Sum of elememts of List2:",s2)
if s1==s2:
 print("Sum of list elements is same")
else:
 print("Sum of list elements is not same")
print("Common elements are:")
for i in range(x):
 for j in range(y):
        if l1[i]==l2[j]:
                print(l1[i])
```

OUTPUT:

```
List 1 [1, 2, 3, 4]
List 2 [1, 3, 2]
Length of lists are different
Sum of elements of List1: 10
Sum of elememts of List2: 6
Sum of list elements is not same
Common elements are:
1
2
3
```

EXPERIMENT 7

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

2) **CODE:**

```
str=input("Enter a string: ")
print("Original string is: ",str)
char=str[0]
str=str.replace(char,'$')
str=char+str[1:]
print("String: ",str)
var=input("Enter a string: ")
 beg=var[0]
 end=var[len(var)-1]
dum=beg
beg=end
end=dum
print(beg+var[1:len(var)-1]+end)
```

OUTPUT:

```
Enter a string: occupation
Original string is:  occupation
String:  occupati$n
Enter a string:
```

EXPERIMENT 8

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

CODE:

```
s=input("Enter a string: ")
t=s[0]
t1=s[-1]
n=len(s)
ns=t1+s[1:n-1]+t
print(ns)
```

OUTPUT:

```
Enter a string: onion
nnioo


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 9

AIM: Accept the radius from user and find area of circle.

CODE:

```
x=input("enter the radious")
x=int(x)
a=3.14*x*x
print(a)
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON$ python3 area.py
enter the radious5
78.5
```

EXPERIMENT 10

AIM: Find biggest of 3 numbers entered**.**

CODE:

```
x=input("Enter the number")
y=input("Enter the number")
z=input("Enter the number")
x=int(x)
y=int(y)
z=int(z)
if(x>y):
        if(x>z):
                print("x is larger")
else:
        if(y>z):
                print("y is larger")
        else:
                print("z is larger")
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON$ python3 greater.py
Enter the number5
Enter the number6
Enter the number1
6 is larger
```

EXPERIMENT 11

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

CODE:

```
import os

a=input("enter the filename : ")

print("The extension of file",a, "is",os.path.splitext(a))
```

OUTPUT:

```
Enter file name:exam.xls
The extension of file exam.xls is ('exam', '.xls')


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 12

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

CODE:

```
colors=[]
str=(input("Enter color names:"))
for i in str.split(','):
 colors.append(i)
print(colors)
print("first color:",colors[0],"Last color:",colors[-1])
```

OUTPUT:

```
Enter color names:red,green,yellow,orenge
['red', 'green', 'yellow', 'orenge']
first color: red Last color: orenge


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 13

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

CODE:

```
n=input("Enter a number :")
nn=n+n
nnn=nn+n
print("The sum is :",int(n)+int(nn)+int(nnn))
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON$ python3 integer.py
Enter a number :5
The sum is : 615
stud@debian:~/Abhinav/PYTHON$
```

EXPERIMENT 14

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

CODE:

```
l1=['red','blue','green','black','yellow']

l2=['orenge','pink','red','brown']

l3=[]

for i in l1:

        if i not in l2:

                l3.append(i)

print(l3)
```

OUTPUT:

```
stud@debian:~/ABHINAV/PYTHON$ python3 color.py
['blue', 'green', 'black', 'yellow']
stud@debian:~/ABHINAV/PYTHON$
```

EXPERIMENT 15

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

CODE:

```
string1="Fisat"

string2="Ankamaly"
f1=string1[0]
f2=string2[0]
string=f2+string1[1:]+" "+f1+string2[1:]
print("The new string is :",string)
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON/python$ gedit swap.py
stud@debian:~/Abhinav/PYTHON/python$ python3 swap.py
The new string is : Aisat Fnkamaly
stud@debian:~/Abhinav/PYTHON/python$
```

EXPERIMENT 16

AIM : Sort dictinary in ascending and descending order.

CODE

```
dict1={"a":1,"c":3,"d":2,"b":4}

l=list(dict1.items())

print(l)

l.sort()

print("Ascending Order is \n",l)
```

```
l=list(dict1.items())

l.sort(reverse=True)

print("Descending order is \n",l)
```

OUTPUT

```
[('a', 1), ('c', 3), ('d', 2), ('b', 4)]
Ascending Order is
 [('a', 1), ('b', 4), ('c', 3), ('d', 2)]
Descending order is
 [('d', 2), ('c', 3), ('b', 4), ('a', 1)]


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 17

AIM: Merge two dictionaries.

CODE:

```
D1={"Name":"Abhinav","Age":"22"}
print("Directory 1",D1)
D2={"Gender":"male","Qualification":"BCA"}
print("Directory 2",D2)
D1.update(D2)
print("After merging...")
print(D1)
```

OUTPUT:

```
Directory 1 {'Name': 'Abhinav', 'Age': '22'}
Directory 2 {'Gender': 'male', 'Qualification': 'BCA'}
After merging...
{'Name': 'Abhinav', 'Age': '22', 'Gender': 'male', 'Qualification': 'BCA'}


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 18

AIM: Find gcd of 2 numbers.

CODE:

```
x=int(input("Enter 1 st number"))

y=int(input("Enter 2nd number"))

if(x>y):

        x1=y

else:

        x1=x

for i in range(1,x1+1):

        if(x%i==0 and y%i==0):

                gcd=i

print("The largest common factor is",gcd)
```

OUTPUT:

```
stud@debian:~/Abhinav/PYTHON$ python3 gcd.py
Enter 1 st number56
Enter 2nd number128
The largest common factor is 8
```

EXPERIMENT 19

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

CODE:

```
l1=[1,2,3,4,5,6,7,8,9,10]
print(l1)
l2=[]
for i in range(len(l1)):
 if l1[i]%2!=0:
        l2.append(l1[i])
print("List after removing even elements")
print(l2)
```

OUTPUT:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
List after removing even elements
[1, 3, 5, 7, 9]


...Program finished with exit code 0
Press ENTER to exit console.
```

COURSE OUTCOME-2

EXPERIMENT 20

AIM: Program to find the factorial of a number

CODE

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

OUTPUT

```
Enter a number:4
1
2
6
24


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 21

AIM: Generate Fibonacci series of N terms

CODE

```
x=input("enter the number")

x=int(x)

f1=0

f2=1

count=0

for i in range(count,x):
```

```
print(f1)

f3=f1+f2

f1=f2

f2=f3
```

OUTPUT

```
stud@debian:~/Abhinav/PYTHON$ python3 fibonacci.py
enter the number5
0
1
1
2
3
stud@debian:~/Abhinav/PYTHON$
```

EXPERIMENT 22

AIM: Find the sum of all items in a list

CODE

```
a=[15,58,66,-99,456,-66,95]

print(sum(a))
```

OUTPUT

```
636


...Program finished with exit code 0
Press ENTER to exit console.
```

EXPERIMENT 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.

CODE:

```
limit1=1000
limit2=9999
list1=[]
for i in range(limit1,limit2):
    j=i
    digit=[]
    while(i!=0):
        digit.append(i%10)
        i=int(i/10)
    count=0
    for n in digit:
        if n%2==0:
            count=count+1
    if count==4:
        for k in range(31,100):
            if((k**2)==j):
                list1.append(j)
                print(k)
print(list1)
```

OUTPUT

```
stud@debian:~$ python3 page.py
68
78
80
92
[4624, 6084, 6400, 8464]
stud@debian:~$
```

EXPERIMENT 24

AIM: Display the given pyramid with the step number accepted from the user.

Eg: N=4

1

2 4

3 6 9

4 8 12 16

CODE:

```
n=int(input("Enter a number:"))
for j in range(0,n+1):
    for i in range(1,j+1):
            i=j*i
            print(i,end=" ")
    print("\n")
```

OUTPUT

```
stud@debian:~$ python3 pyramid.py
Enter a number:4

1

2 4

3 6 9

4 8 12 16
```

EXPERIMENT 25

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

CODE

```
string=input("Enter a string:")
list1=[]
for i in string:
    if i not in list1:
            list1.append(i)
for i in list1:
    count=0
    for j in string:
            if(i==j):
                    count=count+1
    print(i,"\t:",count)
```

OUTPUT

```
stud@debian:~$ python3 frequency.py
Enter a string:welcome
w       : 1
e       : 2
l       : 1
c       : 1
o       : 1
m       : 1
```

EXPERIMENT 26

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

CODE

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

```
print(string)
```

OUTPUT

```
stud@debian:~$ python3 ing.py
Enter a string:Drawing
Drawingly
stud@debian:~$ python3 ing.py
Enter a string:Draw
Drawing
```

EXPERIMENT 27

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

CODE

```
lis=[]
n=int(input("Enter the range:"))
print("Enter the words:")
for i in range(0,n):
        lis.append(input(""))
longest=lis[0]
for i in range(1,n):
        if(len(lis[i])>len(longest)):
          longest=lis[i]
print("Length of longest word is",len(longest))
```

OUTPUT

```
stud@debian:~$ python3 lenlong.py
Enter the range:3
Enter the words:
how
are
you
Length of longest word is 3
```

EXPERIMENT 28

AIM: Construct following pattern using nested loop

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

CODE

```
for i in range(1,6):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
for i in range(4,0,-1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
```

OUTPUT

```
stud@debian:~$ gedit pattern.py
^C
stud@debian:~$ python3 pattern.py
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

EXPERIMENT 29

AIM: Generate all factors of a number**.**

CODE

```
n=int(input("Enter a number:"))
print("Factors are")
for i in range(1,n+1):
   if(n%i==0):
            print(i)
```

OUTPUT

```
stud@debian:~$ python3 fact.py
Enter a number:7
Factors are
1
7
stud@debian:~$ python3 fact.py
Enter a number:12
Factors are
1
2
3
4
6
12
```

COURSE OUTCOME-3

EXPERIMENT 30

Aim: Create a package graphics with modules rectangle, circle and sub-package 3D graphics with module 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.

Terminal(Windows):

mkdir graphics

cd graphics

notepad rectangle

notepad circle

notepad __init__.py

mkdir dgraphics

cd dgraphics

notepad __init__.py

notepad cuboid.py

notepad sphere.py

CODE

1) **Rectangle**

```
class Rectangle:
    def __init__(self,length,width):
        self.length=length
        self.width=width
        def area(self):
            return (self.length*self.width)
        def perimeter(self):
            return (2*(self.length+self.width))
```

2) <u>Circle</u>

```
global pi
pi=3.1416
class Circle:
    global pi
     pi=3.1416
def __init__(self,radius):
     self.radius=radius
def area(self):
    return (pi*(self.radius**2))
def perimeter(self):

    return (2*pi*self.radius)
```

3) <u>Sphere</u>

```
global pi
pi=3.1416
class Sphere:
   def __init__(self,radius):
        self.radius=radius
   def volume(self):
    r=self.radius
 return ((4/3)*pi*(r**3))
  def area(self):
    r=self.radius
return (4*pi*(r**2))
```

4) <u>Cuboid</u>

```
class Cuboid:
  def __init__(self,length,width,height):
     self.l=length
     self.w=width
     self.h=height
def volume(self):
    return (self.l*self.w*self.h)
def area(self):
 #method to find total surface area
   l=self.l
```

```
        w=self.w
        h=self.h
        return (2*((l*w)+(w*h)+(l*h)))
```

CODE

```
from graphics import rectangle as rt

from graphics import circle

from graphics.tdgraphics import *


#Rectangle

r=rt.Rectangle(10,12)

print(&quot;_________________RECTANGLE___________&quot;)

print(&quot;length =&quot;,r.length)

print(&quot;width =&quot;,r.width)

print(&quot;area=&quot;,r.area())

print(&quot;perimeter=&quot;,r.perimeter())


#Circle

c=circle.Circle(12)

print(&quot;_________________CIRCLE_______________&quot;)

print(&quot;radius =&quot;,c.radius)

print(&quot;area=&quot;,c.area())

print(&quot;perimeter=&quot;,c.perimeter())


#Sphere
```

```
s=sphere.Sphere(12)

print(&quot;_________________SPHERE_______________&quot;)

print(&quot;radius =&quot;,s.radius)

print(&quot;area=&quot;,s.area())

print(&quot;volume=&quot;,s.volume())


#Cuboid

cu=cuboid.Cuboid(13,11,14)


print(&quot;_________________CUBOID_______________&quot;)

print(&quot;length =&quot;,cu.l)

print(&quot;width =&quot;,cu.w)

print(&quot;height =&quot;,cu.h)

print(&quot;area=&quot;,cu.area())

print(&quot;volume=&quot;,cu.volume())
```

OUTPUT

```
stud@debian:~$ mkdir graphics
stud@debian:~$ cd graphics
stud@debian:~/graphics$ gedit __init__.py
stud@debian:~/graphics$ gedit rectangle.py
stud@debian:~/graphics$ gedit circle.py
stud@debian:~/graphics$ mkdir tdgraphics
stud@debian:~/graphics$ cd tdgraphics
stud@debian:~/graphics/tdgraphics$ gedit __init__.py
stud@debian:~/graphics/tdgraphics$ gedit __init__.py
stud@debian:~/graphics/tdgraphics$ gedit cuboid.py
stud@debian:~/graphics/tdgraphics$ gedit sphere.py
stud@debian:~/graphics/tdgraphics$ cd ..
stud@debian:~/graphics$ cd ..
stud@debian:~$ gedit graphics.py
stud@debian:~$ python3 graphics.py
```

```
stud@debian:~$ gedit graphics.py
stud@debian:~$ python3 graphics.py
_______________RECTANGLE__________
length = 10
width = 12
area= 120
perimeter= 44
_______________CIRCLE_____________
radius = 12
area= 452.3904
perimeter= 75.3984
_______________SPHERE_____________
radius = 12
area= 1809.5616
volume= 7238.246399999999
_______________CUBOID_____________
length = 13
width = 11
height = 14
area= 958
volume= 2002
stud@debian:~$
```

COURSE OUTCOME-4

EXPERIMENT 31

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

CODE

```
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)
r1=Rectangle(5,8)
r2=Rectangle(8,9)
a1=r1.area()
a2=r2.area()
print("length of r1=",r1.length)
print("breadth of r1=",r1.breadth)
print("length of r2=",r2.length)
print("breadth of r2=",r2.breadth)
print("perimeter of r1=",r1.perimeter())
print("area of r1=",r1.area())
print("perimeterof r2=",r2.perimeter())
print("area of r2=",r2.area())
if(r1.area()>=r2.area()):
        print("area of r1 is largest")
elif(r2.area()>=r1.area()):
        print("area of r2 is largest")
else:
        print("area of r1 and r2 are same")
```

OUTPUT

```
stud@debian:~$ python3 perimeter.py
length of r1= 5
breadth of r1= 8
length of r2= 8
breadth of r2= 9
perimeter of r1= 26
area of r1= 40
perimeterof r2= 34
area of r2= 72
area of r2 is largest
```

EXPERIMENT 32

AIM: Create Bank account with members account number, name, type of account and balance. Write constructor and methods to deposite at the bank and withdraw an amount from the bank.

CODE

```
class Bank_account:
    def __init__(self,ac_no,name,type,balance):
        self.account_number=ac_no
        self.customer_name=name
        self.type_of_account=type
        self.balance=balance
    def deposit(self,amount):
        self.balance=self.balance+amount
    def withdraw(self,amount):
        if(amount>self.balance):
            print("INSUFFICIANT AMOUNT")
        else:
            self.balance=self.balance-amount
account1=Bank_account(100,"Abhinav H","fixed account",100000)
account2=Bank_account(100,"Amalraj","savings account",20000)
account3=Bank_account(103,"Anagha vinayakan","student account",5000)
print("balance of account1 before withdrawal=",account1.balance)
account1.withdraw(1200)
print("balance of account1 after withdrawal=",account1.balance)
print("balance of account 2 before deposit=",account2.balance)
account2.deposit(3400)
print("balance of account 2 after deposit=",account2.balance)
print("balance of account 3 before deposit=",account3.balance)
account3.deposit(4000)
print("balance of account 3 after deposit=",account3.balance)
```

OUTPUT

```
stud@debian:~$ python3 oops.py
balance of account1 before withdrawal= 100000
balance of account1 after withdrawal= 98800
balance of account 2 before deposit= 20000
balance of account 2 after deposit= 23400
balance of account 3 before deposit= 5000
balance of account 3 after deposit= 9000
```

EXPERIMENT 33

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

CODE

```
class Rectangle:

    def __init__(self,l,b):

        self.__length=l

        self.__width=b

    def __lt__(self,ob):


        if((self.__length*self.__width)<(ob.__length*ob.__width)):

            return True

        else:

            return False


r1=Rectangle(15,12)

r2=Rectangle(34,44)

if(r1<r2):
```

```
    print("Area of r1<area of r2")

elif(r2<r1):

    print("Area of r2<area of r1")

else:

    print("Area of r1=area of r2")
```

OUTPUT

```
stud@debian:~$ python3 area.py
Area of r1<area of r2
```

EXPERIMENT 34

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

CODE

```
class Time:
        def __init__(self,h,m,s):
                self.__hour=h
                self.__minute=m
                self.__second=s
        def __add__(self,ob):
                hour=self.__hour+ob.__hour
                minute=self.__minute+ob.__minute
                second=self.__second+ob.__second
                t=Time(hour,minute,second)
                return t

        def print_it(self):
                print("Hour :",self.__hour)
                print("Minute :",self.__minute)
                print("Second :",self.__second)

t1=Time(10,10,10)
t2=Time(20,20,20)
t3=t1+t2
t3.print_it()
```

OUTPUT

```
stud@debian:~$ python3 timer.py
Hour : 30
Minute : 30
Second : 30
```

EXPERIMENT 35

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 number_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overreading.

CODE

```
class Publisher:
        def __init__(self,name):
                self.name=name
                class Book(Publisher):
        def __init__(self,name,title,auther):
                super().__init__(name)
                self.title=title
                self.auther=auther
        def print_function(self):
                print("This Fuction is a member fuction of class Publisher")
class Python(Book):
        def __init__(self,name,title,auther,price,nop):
                super().__init__(name,title,auther)
                self.price=price
                self.nop=nop
        def print_function(self):
                print("Name :",self.name)
                print("Title :",self.title)
                print("Auther :",self.auther)
                print("Price :",self.price)
                print("Number of Pages :",self.nop)
p1=Python("Text book","Python Programming","Mr.abc",100,500)
p1.print_function()
p2=Book("a","b","c")
```

p2.print_function()

OUTPUT

```
stud@debian:~$ gedit publisher.py
stud@debian:~$ python3 publisher.py
Name : Text book
Title : Python Programming
Auther : Mr.abc
Price : 100
Number of Pages : 500
This Fuction is a member fuction of class Publisher
stud@debian:~$
```

COURSE OUTCOME-5

EXPERIMENT 36

Aim : Write a program to read a file line by line and store it into a list

CODE

```
file=open("a.txt","r")
lines=[]
for line in file:
        lines.append(line.strip())
        print(lines)
```

a.txt

On January 15, 2001, Jimmy Wales[6] and Larry Sanger launched Wikipedia; Sanger coined its name as a portmanteau of "wiki" and "encyclopedia."[7][8] Wales was influenced by the "spontaneous order" ideas associated with Friedrich Hayek and the Austrian School of economics, after being exposed to these ideas by Austrian economist and Mises Institute Senior Fellow Mark Thornton.[9] Initially available only in English, versions in other languages were quickly developed. Its combined editions comprise more than 58 million articles, attracting around 2 billion unique device visits per month and more than 17 million edits per month (1.9 edits per second) as of November 2020.[10][11] In 2006, *Time* magazine stated that the policy of allowing anyone to edit had made Wikipedia the "biggest (and perhaps best) encyclopedia in the world."[12]

OUTPUT

```
stud@debian:~/Abhinav/PYTHON$ python3 r.py
['On January 15, 2001,\xa0Jimmy Wales[6]\xa0and\xa0Larry Sanger\xa0launched Wikipedia; Sanger coined its name as a\xa0portmanteau\xa0of "wiki" and "en
cyclopedia."[7][8]\xa0Wales was influenced by the "spontaneous order" ideas associated with\xa0Friedrich Hayek\xa0and the\xa0Austrian School\xa0of eco
nomics, after being exposed to these ideas by Austrian economist and\xa0Mises Institute\xa0Senior Fellow\xa0Mark Thornton.[9]\xa0Initially available o
nly in English, versions in other languages were quickly developed. Its combined editions comprise more than 58 million articles, attracting around 2\
xa0billion unique device visits per month and more than 17 million edits per month (1.9\xa0edits per second) as of November\xa02020.[10][11]\xa0In 200
6,\xa0Time\xa0magazine stated that the policy of allowing anyone to edit had made Wikipedia the "biggest (and perhaps best) encyclopedia in the world.
"[12]']
```

EXPERIMENT 37

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

CODE

```
import csv
with open("text.csv","r") as file:
```

```
reader=csv.reader(file)
    for row in reader:
        print(row)
```

OUTPUT

```
stud@debian:~/Abhinav/PYTHON$ python3 9.py
['Name', 'Batch', 'Roll_no']
['abhinav', 'A', '1']
['ann maria', 'B', '2']
['adheena', 'C', '3']
```