

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-23



20 MCA 132 PROGRAMMING LABORATORY RECORD

Submitted by

HARSHA.B

REG NO: SNG21MCA-2018

in partial fulfillment for the award of the degree in

MASTER OF COMPUTER APPLICATIONS

**SREE NARAYANA GURUKULAM COLLEGE OF
ENGINEERING KADAYIRUPPU, KOLENCHERY 682 311**

(Affiliated to APJ Abdul Kalam Technological University)



20 MCA 132 PROGRAMMING LABORATORY RECORD

*Certified that this is a Bonafide record of practical work done by **Harsha.B** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree in Master of Computer Applications of Sree Narayana Gurukulam College of Engineering done during the Academic year 2021-23.*

Kadayiruppu

Date:

Course Instructor

Head of the Department

Prof.Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2018 on -----

External Examiner

Internal Examiner

SL NO.	DATE	NAME OF EXPERIMENT	PAGE NO.	REMARK
I	CO1			
1	24/11/21	Familiarizing Text Editor, IDE, Code Analysis Tools etc	1	
2	24/11/21	Leap Year	3	
3	24/11/21	List comprehensions	4	
4	24/11/21	occurrences of each word	6	
5	24/11/21	Prompt the user for a list of integers.	7	
6	24/11/21	Store a list of first names.	8	
7	24/11/21	Checking list are of same length,sums to same value,any value occur in both	9	
8	24/11/21	Get a string from an input string and replacing a character	10	
9	24/11/21	Create a string from given string where first and last characters exchanged.	11	
10	24/11/21	Accept the radius from user and find area of circle	12	
11	29/11/21	Find biggest of 3 numbers entered	13	
12	29/11/21	Accept a file name from user and print extension of that	14	
13	29/11/21	Create a list of colors,Display first and last colors.	15	
14	29/11/21	Accept an integer n and compute n+nn+nnn	16	
15	29/11/21	Print out all colors from color-list1 not contained in color-list2	17	
16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	18	
17	29/11/21	Sort dictionary in ascending and descending order	19	
18	29/11/21	Merge two dictionaries	20	
19	29/11/21	Find gcd of 2 numbers.	21	

20	29/11/21	From a list of integers, create a list removing even numbers.	22	
II	CO2			
1	1/12/21	Program to find the factorial of a number	23	
2	1/12/21	Generate Fibonacci series of N terms	24	
3	1/12/21	Find the sum of all items in a list	25	
4	1/12/21	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	26	
5	1/12/21	Display the given pyramid with step number accepted from user	27	
6	1/12/21	Count the number of characters (character frequency) in a string	28	
7	8/12/21	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	29	
8	8/12/21	Accept a list of words and return length of longest word	30	
9	8/12/21	Construct pattern using nested loop	32	
10	8/12/21	Generate all factors of a number. def print_factors(x):	33	
11	8/12/21	Write lambda functions to find area of square, rectangle and triangle.	34	
III	CO3			
1	15/12/21	Work with built-in packages	39	
2	15/12/21	Creation of packages	42	
IV	CO4			
1	9/1/22	Compare two Rectangle objects by their area	43	
2	9/1/22	Create a Bank account with members account number, name, type of account and balance.	46	
3	9/1/22	Overload '<' operator to compare the area of 2 rectangles.	48	

4	9/1/22	Overload '+' operator to find sum of 2 time	50	
5	9/1/22	Use base class constructor invocation and method overriding.	52	
V	CO5			
1	30/1/22	Write a Python program to read a file line by line and store it into a list.	53	
2	30/1/22	Python program to copy odd lines of one file to other	54	
3	30/1/22	Write a Python program to read each row from a given csv file and print a list of strings.	55	
4	30/1/22	Write a Python program to read specific columns of a given CSV file	56	
5	30/1/22	Write a Python program to write a Python dictionary to a csv file.	58	

I. COURSE OUTCOME 1(CO1)

PROGRAM NO: 1

DATE:24/11/2021

AIM: Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE like PyCharm, PyDev...

A text editor is a tool that allows a user to create and revise documents in a computer.

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

An IDE normally consists of at least a source code editor, build automation tools and a debugger

Source code analysis tools, also known as Static Application Security Testing (SAST) Tools, can help analyze source code or compiled versions of code to help find security flaws.

Top Python IDE's

- PyCharm
- Spyder
- Eclipse PyDev
- Wing
- IDLE

PyCharm

In industries most of the professional developers use PyCharm and it has been considered the best IDE for python developers. It was developed by the Czech company JetBrains and it's a cross-platform IDE.

Intelligent code Editor: Smart code editor that facilitates writing high quality Python code.

Availability of integration tool: pycharm provides support from integrating a range of tools.

Integrated Debugging and Testing: Supports for debugging.

Refactoring: Helps in improving the internal structure of python program.

PROGRAM NO: 2**DATE:24/11/2021****AIM:**Display future leap years from current year to a final year entered by user.

```
s=int(input("Enter start year:"))
e=int(input("Enter end year:"))
if (s<e):
    print ("Leap years are:",end=" ")
    for i in range(s,e):
        if i%4==0 and i%100!=0:
            print(i,end=" ")
```

OUTPUT

Enter start year:2021

Enter end year:2050

Leap years are: 2024 2028 2032 2036 2040 2044 2048

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

- **Generate positive list of numbers from a given list of integers**

```
list1=[-10,20,35,-67,70]
re=[num for num in list1 if num>=0]
print(re)
```

OUTPUT

[20, 35, 70]

- **Square of N number**

```
n=int(input("Enter limit:"))
squarelist=[i**2 for i in range(1,n+1)]
print("Square of N numbers:",squarelist)
```

OUTPUT

Enter the limit:5
Result: [1, 4, 9, 16,25]

- **Form a list of vowels selected from a given word**

```
word =str(input("Enter the word:"))
print("The original string is:"+word)
print("The vowel are:",end="")
for i in word:
    if i in 'aeiouAEIOU':
        print([i],end=" ")
```

OUTPUT

Enter the word:python programming
The original string is:python programming
The vowel are:['o'] ['o'] ['a'] ['i']

- **List ordinal value of each element of a word**

```
w=input("Enter a word:")  
print("Ordinal values corresponding to each element is:")  
for i in w:  
    print(i,end=":")  
    print(ord(i),end=" ")
```

OUTPUT

Enter a word:python
Ordinal values corresponding to each element is:
p:112 y:121 t:116 h:104 o:111 n:110

PROGRAM NO: 4**DATE:24/11/2021****AIM:**Count the occurrences of each word in a line of text

```
str1=input ("Enter a string :")  
wordlist =str1.split()  
count =[]  
for w in wordlist:count.append(wordlist.count(w))  
print ("count of the occurrence:"+str(list(zip(wordlist,count))))
```

OUTPUT

```
Enter a string:Python is a programming language  
count of the occurrence:[('Python', 1), ('is', 1), ('a', 1), ('programming', 1),  
( 'language', 1)]
```

PROGRAM NO: 5

DATE:24/11/2021

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

```
n=[]
s=int(input("Enter a limit:"))
print("enter {s} values")
for i in range(0,s):n.append(int(input()))
print("\n the list after assigning:\n")
for i in range(0,len(n)):
    if n[i]>=100:print("over")
    else:print(n[i])
```

OUTPUT

```
Enter a limit:2
Enter {s} values
24
199
```

```
The list after assigning:
24
Over
```

PROGRAM NO: 6

DATE:24/11/2021

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

```
a_list = ["a", "b", "a"]  
occ = a_list.count("a")  
print("count of occurrences of a :",occ)
```

OUTPUT

count of occurrences of a : 2

PROGRAM NO: 7

DATE:24/11/2021

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

```
lst=[1,3,5,7,9,11,34]
lst1=[5,13,45,7,20,65,1]
s=int(0)
c=int(0)
if len(lst)==len(lst1):
    print("Lists are of same length")
else:
    print("Lists have different length")

for i in range(0,len(lst) and len(lst1)):
    s=s+lst[i]
    c=c+lst1[i]
if(s==c):
    print("equal sum")
else:
    print("not same sum")

print("Elements that matched are:")
l=[]
for i in range(0,len(lst)):
    for j in range(0,len(lst1)):
        if lst[i]==lst1[j]:
            l.append(lst[i] and lst1[j])
        else:
            continue
print(l)
```

OUTPUT

Lists are of same length

not same sum

Elements that matched are:

[1, 5, 7]

PROGRAM NO: 8

DATE:24/11/2021

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

```
str1="malayalam"  
char=str1[0]  
str1=str1.replace(char,'$')  
str1=char+str1[1:]  
print(str1)
```

OUTPUT

malayala\$

PROGRAM NO: 9

DATE:24/11/2021

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

```
str = input("Enter a string:")  
new_str = str[-1:] + str[1:-1] + str[:1]  
print("New string : ",new_str)
```

OUTPUT

```
Enter a string: python  
New string: nythop
```


PROGRAM NO: 10

DATE:24/11/2021

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

```
pi=3.14  
r=float(input("Input the radius of the circle:"))  
result=3.14*r**2  
print("The area of the circle with radius is:",result)
```

OUTPUT

Input the radius of the circle:6
The area of the circle with radius is: 113.04

PROGRAM NO: 11**DATE:29/11/2021****AIM:**Find biggest of 3 numbers entered

```
x=int(input("Enter first no:"))
y=int(input("Enter second no:"))
z=int(input("Enter third no:"))
if(x>y) and (x>z):largest=x
elif(y>x) and (y>z):largest=y
else:largest=z
print("The largest no is",largest)
```

OUTPUT

```
Enter first no:56
Enter second no:34
Enter third no:78
The largest no is 78
```

PROGRAM NO: 12

DATE:29/11/2021

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

```
file=input("enter file name:")  
f=file.split(".")  
print("extension of the file is:"+f[-1])
```

OUTPUT

Enter the file name:hello.java

File Name= hello.java

File Extension= java

PROGRAM NO: 13**DATE:29/11/2021**

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

```
a=[]  
for i in range(3):  
    b=input("Enter the color:")  
    a.append(b)  
print(a)  
print("first",a[0])  
print("second",a[2])
```

OUTPUT

```
Enter the color:red  
Enter the color:green  
Enter the color:blue  
['red', 'green', 'blue']  
first red  
second green
```

PROGRAM NO: 14

DATE:29/11/2021

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

```
n=int(input("Enter a number:"))
x=int("%s"%n)
y=int("%s%s"%(n,n))
z=int("%s%s%s"%(n,n,n))
print("n+nn+nnn:",x+y+z)
```

OUTPUT

```
Enter the number:5
n+nn+nnn:615
```

PROGRAM NO: 15

DATE:29/11/2021

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

```
color_list_1=set(["white","pink","red","blue"])  
color_list_2=set(["red","green","pink"])  
print(color_list_1.difference(color_list_2))
```

OUTPUT

```
{'White', 'blue'}
```

PROGRAM NO: 16

DATE:29/11/2021

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

```
a="python"  
b="java"  
p1=a[0]  
p2=b[0]  
c=b[0]+a[1:len(a)]+""+a[0]+b[1:len(b)]  
print(c)
```

OUTPUT

Jythonpava

PROGRAM NO: 17

DATE:29/11/2021

AIM:Sort dictionary in ascending and descending order.

```
import operator
d={1:2,3:3,4:3,2:1,0:0}
print(Original dictionary:,d)
sorted_d=sorted(d.items(),key=operator.itemgetter(1))
print(Dictionary in ascending order by value,sorted_d)
sorted_d=dict(sorted(d.items(),key=operator.itemgetter(1),reverse=True))
print(Dictionary in descending order by value:,sorted_d)
```

OUTPUT

Original dictionary: {1: 2, 3: 3, 4: 3, 2: 1, 0: 0}
Dictionary in ascending order by value [(0, 0), (2, 1), (1, 2), (3, 3), (4, 3)]
Dictionary in descending order by value: {3: 3, 4: 3, 1: 2, 2: 1, 0: 0}

PROGRAM NO: 18

DATE:29/11/2021

AIM:Merge two dictionaries

```
d1 ={'a': 100, 'b': 200}
d2 ={'x' : 300, 'y': 200}
print ("Dict ionary 1=:", d1)
print ("Dictionary 2-: ", d2)
d =d1. copy ()
d.update (d2)
print ("Merged Dictionary: ", d)
```

OUTPUT

Dictionary 1=: {'a': 100, 'b': 200}

Dictionary 2-: {'x': 300, 'y': 200}

Merged Dictionary: {'a': 100, 'b': 200, 'x': 300, 'y': 200}

PROGRAM NO: 19

DATE:29/11/2021

AIM:Find gcd of 2 numbers.

```
n1=int(input("Enter the first number:"))  
  
n2=int(input("Enter the second number:"))  
i=1  
while i<=n1 and i<=n2:  
    if(n1%i==0 and n2%i==0):  
        gcd=i  
        i=i+1  
print("Gcd=",gcd)
```

OUTPUT

Enter the first number:120
Enter the second number:5

Gcd= 5

PROGRAM NO:20**DATE:29/11/2021****AIM:**From a list of integers, create a list removing even numbers.

```
num=[7,8,120,25,44,20,27]
print( "original list:",num)
num=[x for x in num if x%2!=0]
print("list after removing even no:",num)
```

OUTPUT

Original list: [7, 8, 120, 25, 44, 20, 27]

List after removing even no: [7, 25, 27]

II.COURSE OUTCOME 2(CO2)

PROGRAM NO: 1

DATE:1/12/2021

AIM:Program to find the factorial of a number

```
n=int(input("Enter the number:"))
```

```
f=1
```

```
for i in range(1,n+1):
```

```
    f=f*i
```

```
print("Factorial of",n,"=",f)
```

OUTPUT

Enter the number:5

Factorial of 5 is: 120

PROGRAM NO: 2**DATE:1/12/2021****AIM:**Generate Fibonacci series of N terms

```
n=int(input("Enter the limit:"))
a=0
b=1
sum=0
count=1
print("fibonacci series:",end="")
while(count<=n):
    print(sum,end="")
    count+=1
    a=b
    b=sum
    sum=a+b
```

OUTPUT

```
Enter the limit:5
Fibonacci series:
0 1 1 2 3
```

PROGRAM NO: 3

DATE:1/12/2021

AIM:Find the sum of all items in a list

```
list1=[10,15,20,25,30]  
total=sum(list1)  
print("sum of list:",total)
```

OUTPUT

Sum of list:100

PROGRAM NO: 4

DATE:1/12/2021

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

```
from math import sqrt as s
for i in range(1000,10000):
    if s(i)==int(s(i)) and i%2==0:
        print(i,end=" ")
```

OUTPUT

1024 1156 1296 1444 1600 1764 1936 2116 2304 2500 2704 2916 3136 3364
3600 3844 4096 4356 4624 4900 5184 5476 5776 6084 6400 6724 7056 7396 7744
8100 8464 8836 9216 9604

PROGRAM NO:5**DATE:1/12/2021****AIM:**Display the given pyramid with step number accepted from user.

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

OUTPUT

Enter the limit:4

1

2 4

3 6 9

4 8 12 16

PROGRAM NO:6**DATE:1/12/2021****AIM:**Count the number of characters (character frequency) in a string.

```
test_str=str(input("Enter the string:"))
freq={}
for i in test_str:
    if i in freq:
        freq[i]+=1
    else:
        freq[i]=1
print("Count of all characters:"+ str(freq))
```

OUTPUT

Enter the string:malayalam

Count of all characters:{'m': 2, 'a': 4, 'l': 2, 'y': 1}

PROGRAM NO:7

DATE:8/12/2021

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

```
str=input("Enter the string:")
print("Entered string:",str)
if(str.endswith("ing")):
    str=str+"ly"
else:
    str=str+"ing"
print("Resultant string:",str)
```

OUTPUT

Enter the string:play
Entered string: play
Resultant string: playing

PROGRAM NO:8**DATE:8/12/2021****AIM:**Accept a list of words and return length of longest word

```
a=[]
n=int(input("enter the no of elements in list:"))
for x in range(0,n):
    element=input("enter element"+str(x+1))
    a.append(element)
    max1=len(a[0])
    temp=a[0]
for i in a:
    if(len(i)>max1):
        max1=len(i)
        temp=i
print("longest word:",temp)
print("length of longest word:",max1)
```

OUTPUT

```
Enter the no of elements in list:2
Enter element1python
Enter element2programming
Longest word: programming
Length of longest word: 11
```

PROGRAM NO:9

DATE:8/12/2021

AIM:Construct following pattern using nested loop

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

```
n=int(input("Enter the limit:"))  
for i in range(n):  
    for j in range(i):  
        print("*",end=" ")  
    print(" ")  
for i in range(n,0,-1):  
    for j in range(i):  
        print("*",end=" ")  
    print(" ")
```

OUTPUT

Enter the limit:4

*

* *

* * *

* * * *

* * * *

* * *

* *

*

PROGRAM NO:10

DATE:8/12/2021

AIM:Generate all factors of a number. def print_factors(x):

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

OUTPUT

Enter the number:15

Factors of 15 are :

1
3
5
15

PROGRAM NO:11

DATE:8/12/2021

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

```
a_sq=lambda a:a*a
```

```
a_rec=lambda l,b:l*b
```

```
a_tri=lambda b,h:1/2*b*h
```

```
print("Area of square=",a_sq(2))
```

```
print("Area of rectangle=",a_rec(2,2))
```

```
print("Area of triangle=",a_tri(2,5))
```

OUTPUT

Area of square= 4

Area of rectangle= 4

Area of triangle= 5.0

III.COURSE OUTCOME 3(CO3)

PROGRAM NO: 1

DATE:15/12/2021

AIM:Work with built-in packages

Time Module

```
import time
print("current time in sec:",time.time())
print("current time",time.ctime())
print("current time after 30 sec",time.ctime(time.time()+30))
t=time.localtime()
print("time",t)
print("current year",t.tm_year)
print("current month",t.tm_mon)
print("current day",t.tm_wday)
print("current hour",t.tm_hour)
print("current minitue",t.tm_min)
print("current second",t.tm_sec)
```

OUTPUT

current time in sec: 1639964554.0842216

current time Mon Dec 20 07:12:34 2021

current time after 30 sec Mon Dec 20 07:13:04 2021

```
time time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20,
tm_hour=18, tm_min=50,      tm_sec=15, tm_wday=0, tm_yday=354, tm_isdst=0)
current year 2021
current month 12
current day 0
current hour 18
current minitue 50
current second 15
```


Math module

```
import math
print(math.factorial(4))
print (math.gcd(3, 6))
print (math.sqrt(9))
```

OUTPUT

```
24
3
3.0
```

Calendar module

```
import calendar
mm=int(input("Enter month:"))
yy=int(input("Enter year:"))
print("\n")
print(calendar.month(yy,mm))
```

OUTPUT

```
Enter month:1
Enter year:2022
```

```
January 2022
Mo  Tu  We  Th  Fr  Sa  Su
      1   2
 3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
31
```

DateTime module

```
import datetime
t=datetime.time(22,56,20,67)
print(t)
print("Hour",t.hour)
print("Minutes",t.minute)
print("Seconds",t.second)
print("Microsecond:",t.microsecond)
print("\n")
d=datetime.date.today()
print(d)
print("Year:",d.year)
print("Month:",d.month)
print("Day:",d.day)
d1=datetime.date.today()
print(d1)
td=datetime.timedelta(days=2)
print(td)

d2=d1+td
print(d2)
dt=datetime.datetime.combine(d1,t)
print(dt)
```

OUTPUT

```
22:56:20.000067
Hour 22
Minutes 56
Seconds 20
Microsecond: 67
```

```
2021-12-19
Year: 2021
```

Month: 12
Day: 19
2021-12-19
2 days, 0:00:00
2021-12-21
2021-12-19 22:56:20.000067

Statistics module

```
import statistics
l=[4,6,8,9,3,4,5,7,8,7,0,7,3]
a=statistics.mean(l)
print(a)
b=statistics.median(l)
print(b)
c=statistics.mode(l)
print(c)
d=statistics.stdev(l)
print(d)
e=statistics.variance(l)
print(e)
```

OUTPUT

```
5.461538461538462
6
7
2.569545505058064
6.602564102564102
```

Random module

```
import random
l1 = [2, 4, 6, 8, 10, 12]
print(random.choice(l1))
random.seed(4)
print(random.random())
print(random.random())
r1=random.randint(2,5)
print(r1)
```

OUTPUT

```
12
0.23604808973743452
0.1031660342307158
3
```

PROGRAM NO: 2

DATE:15/12/2021

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)

- **graphics package**

circle module

```
def area(r):  
    return(3.14*r*r)
```

```
def perimeter(r):
```

```
return(2*3.14*r)
```

rectangle module

```
def area(l,b):  
    return(l*b)
```

```
def perimeter(l,b):  
    return(2*(l+b))
```

- **3dgraphics package**

sphere module

```
def area(r):  
    return(4*3.14*r*r)
```

```
def perimeter(r):  
    return(2*3.14*r)
```

cuboid module

```
def area(l,w,h):  
    return(2*l*w+2*l*h+2*h*w)
```

```
def perimeter(l,b,h):  
    return(4*(l+b+h))
```

```
from graphics import rectangle  
from graphics import circle  
from dgraphics import cuboid  
from dgraphics import sphere
```

```
print("Rectangle:")  
l=int(input("Enter the length:"))  
b=int(input("Enter the breadth:"))  
print("Area=",rectangle.area(l,b))  
print("Perimeter=",rectangle.perimeter(l,b))
```

```
print("\nCircle:")
r=int(input("Enter the radius:"))
print("Area=",circle.area(r))
print("Perimeter=",circle.perimeter(r))

print("\nCuboid:")

l=int(input("Enter the length:"))
w=int(input("Enter the width:"))
h=int(input("Enter the height:"))
b=int(input("Enter the breadth:"))
print("Area=",cuboid.area(l,w,h))
print("perimeter=",cuboid.perimeter(l,b,h))

print("\nSphere:")
r=int(input("Enter the radius:"))
print("Area=",sphere.area(r))
print("perimeter=",sphere.perimeter(r))
```

OUTPUT

Rectangle:

Enter the length:2

Enter the breadth:2

Area= 4

Perimeter= 8

Circle:

Enter the radius:2

Area= 12.56

Perimeter= 12.56

Cuboid:

Enter the length:2

Enter the width:2

Enter the height:1

Enter the breadth:2

Area= 16

perimeter= 20

Sphere:

Enter the radius:2

Area= 50.24

perimeter= 12.56

IV.COURSE OUTCOME 4(CO4)

PROGRAM NO: 1

DATE:9/1/2022

AIM: 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,breadth,length):
        self.breadth=breadth
        self.length=length
    def area(self):
        return self.breadth*self.length
    def perimeter(self):
        return 2*(self.breadth+self.length)
r1=rectangle(10,20)
r2=rectangle(3,5)
```



```

print("Area of rectangle 1 :",r1.area())
print("Area of rectangle 2 :",r2.area())
print("Peimeter of rectangle 1:",r1.perimeter())
print("Peimeter of rectangle 2:",r2.perimeter())
if(r1.area()>r2.area()):
    print("recangle 1 is of greater area")
else:
    print("recangle 2 is of greater area")

```

OUTPUT

```

Area of rectangle 1 :200
Area of rectangle 2 ; 15
Perimeter of rectangle 1 :60
Perimeter of rectangle 2 :16
rectangle 1 is of greater area

```

PROGRAM NO: 2

DATE:9/1/2022

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.

```

class bank:
    bal=0
    def __init__(self,accno,name,ac_type,bal):
        self.accno=accno
        self.name=name
        self.ac_type=ac_type
        self.bal=bal

    def display(self):
        print("\nAccount Info:")

```

```

    print("Account Number:",self.accno)
    print("Account Name:",self.name)
    print("Account Type:",self.ac_type)
    print("Account Balance:",self.bal)

def deposit(self):
    dep=int(input("Enter the amount to deposit:"))
    self.bal=self.bal+dep

def withdraw(self):
    w=int(input("Enter the amount to withdraw:"))
    if w > self.bal:
        print("Insufficient Balance")
    else:
        self.bal=self.bal-w
        print("RS-",w,"Withdrawn successfully")

acc_no=int(input("Enter the Account Number:"))
acc_name=input("Enter the name:")
acc_type=input("Enter the account type-(savings/current):")
balance=int(input("Enter the initial balance:"))
b1=bank(acc_no,acc_name,acc_type,balance)

while(1):
    print("\n1.Account Info\n2.Deposit\n3.Withdraw\n4.Exit")
    opt=int(input("Select your option:"))
    if opt == 1:
        b1.display()
    elif opt == 2:
        b1.deposit()
    elif opt == 3:
        b1.withdraw()
    elif opt == 4:
        print("Exited")
        break
    else:
        print("Invalid Option")

```

OUTPUT

Enter the Account Number:1654

Enter the name:Harsha

Enter the account type-(savings/current):savings

Enter the initial balance:2000

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:2

Enter the amount to deposit:2000

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account Info:

Account Number: 1654

Account Name: Harsha

Account Type: savings

Account Balance: 4000

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:3

Enter the amount to withdraw:1000

RS- 1000 Withdrawn successfully

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account Info:

Account Number: 1654

Account Name: Harsha

Account Type: savings

Account Balance: 3000

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:4

Exited

PROGRAM NO: 3

DATE:9/1/2022

AIM: 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,b):
```

```
    self.__length=l
```

```
    self.__breadth=b
```

```
def area(self):
```

```
    self.area=self.__length*self.__breadth
```

```
    print("Area=",self.area)
```

```

def __lt__(self,second):
    if self.area < second.area:
        return True
    else:
        return False

print("first Rectangle:")
len1=int(input("Enter the length:"))
bread1=int(input("Enter the breadth:"))
obj1=rectangle(len1,bread1)
obj1.area()

print("\nSecond Rectangle:")
len2=int(input("Enter the length:"))
bread2=int(input("Enter the breadth:"))
obj2=rectangle(len2,bread2)
obj2.area()
if obj1 < obj2 :
    print("\nArea of second rectangle is larger:")
else:
    print("\nArea of first rectangle is larger:")

```

OUTPUT

first Rectangle:
 Enter the length:4
 Enter the breadth:3
 Area= 12

Second Rectangle:
 Enter the length:2
 Enter the breadth:2
 Area= 4

Area of first rectangle is larger:

PROGRAM NO: 4

DATE:9/1/2022

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

```
class Time:
    def __init__(self, hour, minute, second):
        self.__hour = hour
        self.__minute = minute
        self.__second = second
    def __add__(self, a2):
        second = self.__second + a2.__second
        minute = self.__minute + a2.__minute
        hour = self.__hour + a2.__hour
        if (second > 60):
```

```

        second=second-60
        minute=minute+1
    if(minute>60):
        minute=minute-60
        hour=hour+1
    return hour,minute,second
print("Enter time1:")
h1=int(input("Hour:"))
m1=int(input("Minute:"))
s1=int(input("Second:"))

t1=Time(h1,m1,s1)

print("Enter time2:")
h2=int(input("Hour:"))
m2=int(input("Minute:"))
s2=int(input("Second:"))

t2=Time(h2,m2,s2)

hr,min,sec=t1+t2
print(hr,end=":")
print(min,end=":")
print(sec,end=" ")

```

OUTPUT

```

Enter time1:
Hour:12
Minute:27
Second:38
Enter time2:
Hour:11
Minute:45
Second:28
24:13:1

```

PROGRAM NO:5**DATE:9/1/2022****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.

```
class publisher:

    def __init__(self,pname):
        self.pname=pname

    def display(self):
        print("Publisher Name:",self.pname)
```



```
class book(publisher):

    def get(self,title,author):
        self.title=title
        self.author=author

    def display(self):
        print("Title Name:",self.title)
        print("Author Name:",self.author)

class python(book):

    def __init__(self,price,nop,pname):

        super().__init__(pname)
        self.price=price
        self.nop=nop

    def details(self):
        print("Price:",self.price)
        print("No of pages:",self.nop)

s1=python(450,72,"K D")
s1.get("Flames","K D")
s1.display()
s1.details()
```

OUTPUT

Title Name: The Indian Story
Author Name: Bimal Jalan
Price: 500
No of pages: 100

V.COURSE OUTCOME 5(CO5)

PROGRAM NO: 1

DATE:30/1/2022

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

```
f1=open("sample.txt","w")
f1.write("This is my first line.\n This is my second line \n This is my third line")
f1=open("sample.txt","r")
ff=f1.readlines()
print(ff)
```

OUTPUT

```
['This is my first line.\n', ' This is my second line \n', ' This is my third line']
```

This is my first line.
This is my second line
This is my third line

PROGRAM NO: 2

DATE:30/1/2022

AIM:Python program to copy odd lines of one file to other

```
f1=open("sample.txt","r")
ff=f1.readlines()
with open("odd.txt","w") as f2:
    for x in range(0,len(ff)):
        if(x%2!=0):
            f2.write(ff[x])
```

OUTPUT

This is my second line

PROGRAM NO: 3

DATE:30/1/2022

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

```
import csv
with open('departments.csv', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=' ', quotechar='|')
    for r in data:
        print(' '.join(r))
```

department.csv

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
```

30,Purchasing,114,1700

OUTPUT

department_id,department_name,manager_id,location_id

10,Administration,200,1700

20,Marketing,201,1800

30,Purchasing,114,1700

PROGRAM NO: 4

DATE:30/1/2022

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

```
import csv
with open('departments.csv', newline='') as csvfile:
    data = csv.DictReader(csvfile)
    print("id Department")
    print("-----")
    for r in data:
        print(r['department_id'], " ", r['department_name'])
```

department.csv

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
30,Purchasing,114,1700
```

OUTPUT

```
id Department
-----
```

```
10 Administration
20 Marketing
30 Purchasing
```

PROGRAM NO: 5

DATE:30/1/2022

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.

```
import csv
```

```
field_names = ['No', 'Company', 'Model']
```

```
cars = [
    {'No': 1, 'Company': 'Ferrari', 'Model': '488 GTB'},
    {'No': 2, 'Company': 'Porsche', 'Model': '918 Spyder'},
    {'No': 3, 'Company': 'Bugatti', 'Model': 'La Voiture Noire'},
]
```

```
with open('cars.csv', 'w') as csvfile:
```

```
writer = csv.DictWriter(csvfile, fieldnames=field_names)
writer.writeheader()
writer.writerows(cars)
```

```
with open('cars.csv', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=' ', quotechar='|')
    for r in data:
        print(' '.join(r))
```

cars.csv

No,Company,Model

1,Ferrari,488 GTB

2,Porsche,918 Spyder

3,Bugatti,La Voiture Noire

OUTPUT

No,Company,Model

1,Ferrari,488, GTB

2,Porsche,918, Spyder

3,Bugatti,La, Voiture, Noire

