SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-22

20 MCA 132 PROGRAMMING LAB LABORATORY RECORD

Submitted by

SISIRA K

REG NO: SNG21MCA-2034

in partial fulfillment for the award of Master’s degree in

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 doneby ***SISIRA K*** *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 2021-22.*

Kadayiruppu Course Instructor

Date: ……………….

Head of the Department

Prof.Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2034 on ------------

External Examiner InternalExaminer

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	2	
3	24/11/21	List comprehensions	3	
4	24/11/21	occurrences of each word	5	
5	24/11/21	Prompt the user for a list of integer.	6	
6	24/11/21	Store a list of first names.	7	
7	24/11/21	Checking list are of same length,sums to same value,any value occur in both	8	
8	24/11/21	Get a string from an input string and replacing a character	9	
9	24/11/21	Create a string from given string where first and last characters exchanged.	10	
10	24/11/21	Accept the radius from user and find area of circle	11	
11	29/11/21	Find biggest of 3 numbers entered	12	
12	29/11/21	Accept a file name from user and print extension of that	13	
13	29/11/21	Create a list of colors,Display first and lastcolors.	14	
14	29/11/21	Accept an integer n and compute n+nn+nnn	15	
15	29/11/21	Print out all colors from color-list1 not contained in color-list2	16	
16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	17	
17	29/11/21	Sort dictionary in ascending and descending order	18	

18	29/11/21	Merge two dictionaries	19	
19	29/11/21	Find gcd of 2 numbers.	20	
20	29/11/21	From a list of integers, create a list removing even numbers.	21	
II	CO2			
1	1/12/21	Program to find the factorial of a number	22	
2	1/12/21	Generate Fibonacci series of N terms	23	
3	1/12/21	Find the sum of all items in a list	24	
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.	25	
5	1/12/21	Display the given pyramid with step number accepted from user	26	
6	1/12/21	Count the number of characters (character frequency) in a string	27	
7	8/12/21	Add‘ing’attheendofagivenstring.Ifitalreadyendswith‘ing’,thenadd‘ly’	28	
8	8/12/21	Accept a list of words and return length of longest word	29	
9	8/12/21	Construct pattern using nested loop	30	
10	8/12/21	Generate all factors of a number. def print_factors(x):	32	
11	8/12/21	Write lambda functions to find area of square, rectangle and triangle.	33	
III	CO3			
1	15/12/21	Work with built-in packages	34	
2	15/12/21	Creation of packages	38	
IV	CO4			
1	9/1/22	Compare two Rectangle objects by their area	41	

2	9/1/22	Create a Bank account with members account number, name, type of account and balance.	42	
3	9/1/22	Overload ‘<’ operator to compare the area of 2 rectangles.	43	
4	9/1/22	Overload ‘+’ operator to find sum of 2 time	45	
5	9/1/22	Use base class constructor invocation and method overriding.	47	
V	CO5			
1	30/1/22	Write a Python program to read a file line by line and store it into a list.	49	
2	30/1/22	Python program to copy odd lines of one file to other	50	
3	30/1/22	Write a Python program to read each row from a given csv file and print a list of strings.	51	
4	30/1/22	Write a Python program to read specific columns of a given CSV file	52	
5	30/1/22	Write a Python program to write a Python dictionary to a csv file.	53	

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...

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:
An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
Build, execution, and debugging tools
Some form of source control
Most IDEs support many different programming languages and contain many more features. They can, therefore, be large and take time to download and install. You may also need advanced knowledge to use them properly.

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.

- It is considered as an intelligent code editor, fast and safe refactoring, and smart code.
- Features for debugging, profiling, remote development, testing the code, auto code completion, quick fixing, error detection and tools of the database.
- Support for Popular web technologies, web frameworks, scientific libraries and version control.

PROGRAM NO: 2

DATE:24/11/2021

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

PROGRAM

```
s=int(input("enter start year:"))
e=int(input("enter end year:"))
if(s<e):
print("leap year is",end=" ")
fori in range(s,e):
if i%4==0 and i%100!=0:
print(i,end=" ")
```

OUTPUT

```
enter start year:2031
enter end year:2070
leap year is 2032 2036 2040 2044 2048 2052 2056 2060 2064 2068
```

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

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

PROGRAM

```
list1=[2,3,-4,-5,-7,8]
list2=[]
for i in list1:
if i>0:
list2.append(i)
print("Resultant list",list2)
```

OUTPUT

```
Resultant list [2, 3, 8]
```

- Square of   Number

PROGRAM

```
n=int(input("Enter the limit:"))
list1=[]
sq=1
for i in range(1,n+1):
sq=i*i
list1.append(sq)
print("Result:",list1)
```

OUTPUT

```
enter the limit:5
sqaure [1, 4, 9, 16, 25]
```

- Form a list of vowels selected from a given word

PROGRAM

```
word=str(input("enter the string:"))
      print("the actual string is",word)
print("vowels are:",end=" ")
fori in word:
ifi in "aeiou,AEIOU":
print(i,end=" ")
```

OUTPUT

```
enter the string:HUMMING BIRD
the actual string is HUMMING BIRD
vowels are: U I I
```

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

PROGRAM

```
w1=input("Enter the word:")
for i in w1:
print(i," ")
print(ord(i),"\n")
```

OUTPUT

```
Enter the word:hai
h
104
a
97
i
105
```

PROGRAM NO: 4

DATE:24/11/2021

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

PROGRAM

```
str1=str(input("enter the string:"))

wordlist=str1.split()

count=[]

for w in wordlist:

count.append(wordlist.count(w))

print("count of the occurence:",str(list(zip(wordlist,count))))
```

OUTPUT

```
enter the string:MALAYALAM
count of the occurence: [('MALAYALAM', 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.

PROGRAM

```
n=[]
s=int(input("Enter a limit:"))
print("Enter {s} values")
fori in range(0,s):n.append(int(input()))
print("\nThe list after assinging:\n")
fori 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 assinging:

24
over
```

PROGRAM NO: 6

DATE:24/11/2021

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

PROGRAM

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

OUTPUT

```
Count of   occurrence 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

PROGRAM

```
lst=[1,3,5,7,9,11,34]
lst1=[5,13,45,7,20,65,1]
s=int(0)
c=int(0)

iflen(lst)==len(lst1):
print("Lists are of same length")
else:
print("Lists have different length")

fori 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=[]
fori in range(0,len(lst)):
for j in range(0,len(lst1)):
iflst[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]

PROGRAM

```
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.

PROGRAM

```
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.

PROGRAM

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

OUTPUT

```
input the radius:4
the area of the circle with radius is: 50.24
```

PROGRAM NO: 11

DATE:29/11/2021

AIM: Find biggest of 3 numbers entered

PROGRAM

```
x = int(input("Enter 1st number: "))
y = int(input("Enter 2nd number: "))
z = int(input("Enter 3rd number: "))
if (x > y) and (x > z):
largest = x
elif (y > x) and (y > z):
largest = y
else:
largest = z
print("The largest number is",largest)
```

OUTPUT

```
Enter 1st number: 54
Enter 2nd number: 67
Enter 3rd number: 32
The largest number is 67
```

PROGRAM NO: 12

DATE:29/11/2021

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

PROGRAM

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

OUTPUT

```
Enter file name:SISIRA.JAVA
extension of    the file is: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 lastcolors

PROGRAM

```
a=[]
fori in range(3):
        b=input("enter the color:")
a.append(b)
print(a)
print(a[0])
print(a[2])
```

OUTPUT

```
enter the color:RED
enter the color:BLUE
enter the color:GREEN
['RED', 'BLUE', 'GREEN']
RED
GREEN
```

PROGRAM NO: 14

DATE:29/11/2021

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

PROGRAM

```
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 a number:2
n+nn+nnn: 246
```

PROGRAM NO: 15

DATE:29/11/2021

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

PROGRAM

```
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.

PROGRAM

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

OUTPUT

```
hitekat
```

PROGRAM NO: 17

DATE:29/11/2021

AIM: Sort dictionary in ascending and descending order.

PROGRAM

```
import operator
d = {1: 2, 3: 4, 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: 4, 4: 3, 2: 1, 0: 0}
Dictionary in ascending order by value    [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]
Dictionary in descending order by value :    {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
```

PROGRAM NO: 18

DATE:29/11/2021

AIM: Merge two dictionaries

PROGRAM

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

OUTPUT

```
Dictionary 1= {'a': 50, 'b': 150}
Dictionary 2= {'x': 250, 'y': 200}
Merged Dictionary:    {'a': 50, 'b': 150, 'x': 250, 'y': 200}
```

PROGRAM NO: 19

DATE:29/11/2021

AIM: Find gcd of 2 numbers.

PROGRAM

```
x= int(input("Enter 1st number: "))
y= int(input("Enter 2nd number: "))
i = 1
while(i<= x and i<= y):
if(x % i == 0 and y% i == 0):
gcd = i
i = i + 1
print("GCD :", gcd)
```

OUTPUT

```
Enter 1st number: 34
Enter 2nd number: 65
GCD : 1
```

PROGRAM NO: 20

DATE:29/11/2021

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

PROGRAM

```
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 numbers:",num)
```

OUTPUT

```
Original list: [7, 8, 120, 25, 44, 20, 27]
list after removing Even numbers: [7, 25, 27]
```

II. COURSE OUTCOME 2(CO2)

PROGRAM NO: 1

DATE:1/12/2021

AIM: Program to find the factorial of a number

PROGRAM

```
n=int(input("enter the number"))
f=1
fori in range(1,n+1):
        f=f*i
print("factorial of",n,"=",f)
```

OUTPUT

```
enter the number7
factorial of 7 = 5040
```

PROGRAM NO: 2

DATE:1/12/2021

AIM: Generate Fibonacci series of N terms

PROGRAM

```
n=int(input("enter the number"))
a=0
b=1
sum=0
count=1
print("fibonacciseries",end=" ")
while(count<=n):
print(sum,end=" ")
count+=1
    a=b
    b=sum
sum=a+b
```

OUTPUT

```
enter the number6
fibonacci series 0 1 1 2 3 5
```

PROGRAM NO: 3

DATE:1/12/2021

AIM: Find the sum of all items in a list

PROGRAM

```
list1 = [10, 15, 20, 25, 30]

total = sum(list1)
```

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.

```
import math
n=int(input("Enter the limit:"))
print("Perfect numbers:\n")
for i in range(1000,n+1):
  t=int(math.sqrt(i))
if t*t==i and i%2==0:
    print(i,end="")
```

OUTPUT

```
Enter the limit:1200
Perfect numbers:
1024 1156
```

PROGRAM NO: 5

DATE:1/12/2021

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

PROGRAM

```
rows = int(input("Enter the number of rows: "))
fori in range(1, rows+1):
for j in range(1,i+1):
print(i * j, end=' ')
print()
```

OUTPUT

```
Enter the number of rows: 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.

PROGRAM

```
str1=input("Enter the string:")
f={}
for i in str1:
if i in f:
f[i]=f[i]+1
else:
f[i]=1
print(f)
```

OUTPUT

```
Enter the string:messi
{'m': 1, 'e': 1, 's': 2, 'i': 1}
```

PROGRAM NO: 7

DATE:8/12/2021

AIM: Add‘ing’attheendofagivenstring.Ifitalreadyendswith‘ing’,thenadd‘ly’

PROGRAM

```
str=input("enter a string:")
print("print the string:",str)
if(str.endswith("ing")):
str=str+'ly'
else:
str=str+'ing'
print("print new string",str)
```

OUTPUT

```
enter a string:play
print the string: play
print new string playing


enter a string:coming
print the string: coming
print new string comingly
```

PROGRAM NO: 8

DATE:8/12/2021

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

PROGRAM

```
a=[]
n=int(input("Enter the number 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]
fori in a:
if(len(i)>max1):
        max1=len(i)
temp=i
print("Longest Word:",temp)
print("Length of longest word :",max1)
```

OUTPUT

```
Enter the number of elements in list:2
Enter element 1fantastic
Enter element 2performance
Longest Word: performance
Length of longest word : 11
```

PROGRAM NO:9

DATE:8/12/2021

AIM: Construct following pattern using nested loop

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

PROGRAM

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

PROGRAM

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

OUTPUT

```
Enter the number:16
Factors of 16 :
1
2
4
8
16
```

PROGRAM NO: 11

DATE:8/12/2021

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

PROGRAM

```
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

<u>Time Module</u>

PROGRAM

```
import time
print("Current time in sec:",time.time())
print("Current time:",time.ctime())
print("Time After 30 sec:",time.ctime(time.time()+30))
t=time.localtime()
print("Time:",t)
print("Time-current year:",t.tm_year)
print("Time:-current month",t.tm_mon)
print("Time:-current day",t.tm_mday)
print("Time:-current hour",t.tm_hour)
print("Time:-current minute",t.tm_min)
print("Time:-current sec",t.tm_sec)
print("Time:-current week day",t.tm_wday)
print("Time:-current year day",t.tm_yday)
```

OUTPUT

```
Current time in sec: 1639915265.630671
Current time: Sun Dec 19 17:31:05 2021
Time After 30 sec: Sun Dec 19 17:31:35 2021
Time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=19,
tm_hour=17, tm_min=31, tm_sec=5, tm_wday=6, tm_yday=353, tm_isdst=0)
Time-current year: 2021
Time:-current month 12
Time:-current day 19
Time:-current hour 17
Time:-current minute 31
```

```
Time:-current sec 5
Time:-current week day 6
Time:-current year day 353
```

Math module

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

OUTPUT

```
24
3
3.0
```

Calendar module

PROGRAM

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

OUTPUT

```
Enter month: 2
Enter year :2000
  February 2000

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
```

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

PROGRAM

```
import statistics
l1=[1,2,3,4,5]
print(statistics.mean(l1))
print(statistics.mean([1, 3, 5, 7, 9, 11])) #mean
        print(statistics.harmonic_mean([40, 60, 80]))
print(statistics.harmonic_mean([10, 30, 50, 70, 90]))
```

OUTPUT

```
3
6
55.38461538461538
27.97513321492007
```

Random

PROGRAM

```
import random
mylist = ["apple", "banana", "cherry"]
print(random.choice(mylist))
print(random.choices(mylist, k=2))
print(random.sample(mylist, k=2))
random.shuffle(mylist)
print(mylist)
print(random.randrange(3, 9))
```

OUTPUT

```
0.5714025946899135
['banana', 'cherry']
0.5780913011344704
['cherry', 'banana', 'apple']
banana
```

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 andperimeter of respective figures in each module. Write programs that finds areaand perimeter of figures by different importing statements. (Include selectiveimport of modules and import * statements)

PROGRAM

- **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 methodsto find area and perimeter. Compare two Rectangle objects by their area.

PROGRAM

```
class rectangle():
def __init__(self,breadth,length):
self.breadth=breadth
self.length=length
def area(self):
returnself.breadth*self.length
def perimeter(self):
return 2*(self.breadth+self.length)
r1=rectangle(45,24)
r2=rectangle(10,20)
print("Area of rectangle 1 :",r1.area())
print("Area of rectangle 2 :",r2.area())
print("perimeter of rectangle 1:",r1.perimeter())
print("perimeter 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 : 1080
Area of rectangle 2 : 200
perimeter of rectangle 1: 138
perimeter of rectangle 2: 60
recangle 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.

PROGRAM

```
class Bank:
def __init__(self):
self.bal=0
print("account is created")
def deposit(self):
amount=int(input("enter amount to deposit"))
self.bal=self.bal+amount
print("balance:",self.bal)
def withdraw(self):
amount=int(input("enter amount to withdraw"))
if(amount>self.bal):
print("Insufficient Balance!")
else:
self.bal=self.bal-amount
print("Your Remaining Balance=",self.bal)
def enquiry(self):
print("Your Balance =",self.bal)
b1= Bank()
b1.deposit()
b1.withdraw()
b1.enquiry()
```

OUTPUT

```
account is created
enter amount to deposit20000
balance: 20000
enter amount to withdraw200
Your Remaining Balance= 19800
Your Balance = 19800
```

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.

PROGRAM

```
class rectangle:
        __area = 0
        __perimeter = 0
def __init__(self,length,breadth):
self.__length = length
self.__breadth = breadth
defcalc_area(self):
self.__area = self.__length*self.__breadth
print("Area is    :",self.__area)
def __lt__(self,second):
ifself.__area<second.__area:
return True
else:
return False
length1= int(input("Enter length of the rectangle 1 : "))
breadth1 = int(input("Enter width of the rectangle    1 : "))
length2 = int(input("Enter length of the rectangle    2    : "))
breadth2 = int(input("Enter width of the rectangle    2 : "))
obj1 = rectangle(length1,breadth1)
obj2 = rectangle(length2,breadth2)
obj1.calc_area()
obj2.calc_area()
if obj1 < obj2:
print("Rectangle two is large")
else:
print("Rectangle one is large or these are equal")
```

OUTPUT

```
Enter length of the rectangle 1 : 3
Enter width of the rectangle    1 : 4
Enter length of the rectangle    2    : 46
Enter width of the rectangle    2 : 5
Area is    : 12
Area is    : 230
Rectangle two is large
```

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.

PROGRAM

```
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
returnhour,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:2
minute:3
second3
Enter time2:
hour:4
minute:2
second43
6:5:46
```

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

PROGRAM

```
class publisher:
def __init__(self,title,author):
self.title=title
self.author=author
def display(self):
print("Title:",self.title)
print("Author:",self.author)
class book(publisher):
def __init__(self,price,no_of_page):
self.price=price
self.no_of_page=no_of_page
def display(self):
print("Price:",self.price)
print("No. of Pages:",self.no_of_page)
class python(book):
def __init__(self,title,author,price,no_of_page):
        publisher.__init__(self,title,author)
        book.__init__(self,price,no_of_page)
def display(self):
      print("Title:",self.title)
      print("Author:",self.author)
      print("Price:",self.price)
      print("No. of Pages:",self.no_of_page)
      p=python("Python Programming","AnilKumar",1000,120)
      p.display()
```

OUTPUT

```
Title: Python Programming
Author: AnilKumar
Price: 1000
No. of Pages: 120
```

V. COURSE OUTCOME 5(CO5)

PROGRAM NO: 1

DATE:30/1/2022

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

PROGRAM

```
f1=open("myfile.txt","w")
f1.write("This is my first file in python.\nWant to work with files.\nThis is my third")
f1.close()
f1=open("myfile.txt","r")
f1.seek(0,0)
ff=f1.readlines()
for x in range(0,len(ff)):
        print(ff[x])
print()
print(ff)
f1.close()
```

OUTPUT

```
This is my first file in python.
Want to work with files.
This is my third
['This is my first file in python.\n', 'Want to work with files.\n', 'This is my third']
```

PROGRAM NO: 2

DATE:30/1/2022

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

PROGRAM

```
f1=open("myfile.txt","r")
for x in f1:
        print(x)

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

OUTPUT

```
This is my first file in python.
Want to work with files.
This is my third
Want to work with files.
```

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.

PROGRAM

```
import csv
filename = "username.csv"
rows = []
cf=open(filename, 'r')
csvreader = csv.reader(cf)
for r in csvreader:
    rows.append(r)
print(rows)
cf.close()
```

OUTPUT

```
[['Username;Identifier;Firstname;Lastname'],
['booker12;9012;Rachel;Booker'],['grey07;2070;Laura;Grey'],
['johnson81;4081;Craig;Johnson'],['jenkins46;9346;Mary;Jenkins'],
['smith79;5079;Jamie;Smith']]
```

PROGRAM NO: 4

DATE:30/1/2022

AIM: Write a python program to read specific columns of a given cvs file and print the content of the columns.

PROGRAM

```
import csv
filename = "emp.txt"
fields = []
rows = []
cf=open(filename, 'r')
csvreader = csv.DictReader(cf)
for r in csvreader:
    print(dict(r))
```

OUTPUT

```
{'gname': 'John Smith', 'department': 'Accounting', 'birthday month': 'November'}
{'gname': 'Erica Meyers', 'department': 'IT', 'birthday month': 'March'}
```

PROGRAM NO: 5

DATE:30/1/2022

AIM: Write a python program to write a python dictionary to cvs.file.After writing the cvs file read the cvs file and display the content.

PROGRAM

```
import csv
field_names = ['No', 'Company', 'Car Model']
cars = [
{'No': 1, 'Company': 'Ferrari', 'Car Model': '488 GTB'},
{'No': 2, 'Company': 'Porsche', 'Car Model': '918 Spyder'},
{'No': 3, 'Company': 'Bugatti', 'Car Model': 'La Voiture Noire'},
{'No': 4, 'Company': 'Rolls Royce', 'Car Model': 'Phantom'},
{'No': 5, 'Company': 'BMW', 'Car Model': 'BMW X7'},
]
with open('Names1.csv', 'w') as csvfile:
        writer = csv.DictWriter(csvfile, fieldnames = field_names)
        writer.writeheader()
        writer.writerows(cars)
filename = "names1.csv"
cf=open("names1.csv", 'r')
rows=[]
csvreader = csv.reader(cf)
for r in csvreader:
    rows.append(r)
for r in rows:
            print(*r)
```

OUTPUT

```
No Company Car Model
1 Ferrari 488 GTB
2 Porsche 918 Spyder
3 Bugatti La Voiture Noire
4 Rolls Royce Phantom
5 BMW BMW X7
```