

# **SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING**

**KADAYIRUPPU, KOLENCHERY 682311**

**(Affiliated to APJ Abdul Kalam Technological University)**

**ACADEMIC YEAR 2021-23**



## **PROGRAMMING LAB LABORATORY RECORD**

*Submitted by*

**ANANDHU RAJ**

**REG NO: SNG21MCA-2007**

*in partial fulfilment 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)**



**PROGRAMMING LAB LABORATORY RECORD**

*Certified that this is a Bonafide record of practical work done by **ANANDHU RAJ** 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

Course Instructor:

Date:.....

Head of the Department

**Prof.Dr.SANDHYA R**

Submitted for University Practical Examination

**Reg No.SNG21MCA-2007** 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	11	
9	24/11/21	Create a string from given string where first and last characters exchanged.	12	
10	24/11/21	Accept the radius from user and find area of circle	13	
11	29/11/21	Find biggest of 3 numbers entered	14	
12	29/11/21	Accept a file name from user and print extension of that	15	
13	29/11/21	Create a list of colors,Display first and last colors.	16	
14	29/11/21	Accept an integer n and compute n+nn+nnn	17	
15	29/11/21	Print out all colors from color-list1 not contained in color-list2	18	

16	29/11/21	Create a single string separated with space from two strings by swapping the character at position 1.	19	
17	29/12/21	Sort dictionary in ascending and descending order	20	
18	29/12/21	Merge two dictionaries	21	
19	29/11/21	Find gcd of 2 numbers.	22	
20	29/11/21	From a list of integers, create a list removing even numbers.	23	
II	CO2			
1	2/12/21	Program to find the factorial of a number	24	
2	2/12/21	Generate Fibonacci series of N terms	25	
3	2/12/21	Find the sum of all items in a list	26	
4	2/12/21	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	27	
5	2/12/21	Display the given pyramid with step number accepted from user	28	
6	2/12/21	Count the number of characters (character frequency) in a string	29	
7	2/12/21	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'	30	
8	2/12/21	Accept a list of words and return length of longest word	31	
9	2/12/21	Construct pattern using nested loop	32	
10	8/12/21	Generate all factors of a number. def print_factors(x):	34	
11	8/12/21	Write lambda functions to find area of square, rectangle and triangle.	35	
III	CO3			
1	20/12/21	Work with built-in packages	36	

<b>2</b>	<b>20/12/21</b>	<b>Creation of packages</b>	<b>41</b>	
<b>IV</b>	<b>CO4</b>			
<b>1</b>	<b>11/1/22</b>	<b>Compare two Rectangle objects by their area</b>	<b>44</b>	
<b>2</b>	<b>11/1/22</b>	<b>Create a Bank account with members account number, name, type of account and balance.</b>	<b>46</b>	
<b>3</b>	<b>11/1/22</b>	<b>Overload '&lt;' operator to compare the area of 2 rectangles.</b>	<b>50</b>	
<b>4</b>	<b>11/1/22</b>	<b>Overload '+' operator to find sum of 2 time</b>	<b>52</b>	
<b>5</b>	<b>12/1/22</b>	<b>Use base class constructor invocation and method overriding.</b>	<b>54</b>	
<b>V</b>	<b>CO5</b>			
<b>1</b>	<b>19/1/22</b>	<b>Write a Python program to read a file line by line and store it into a list.</b>	<b>56</b>	
<b>2</b>	<b>19/1/22</b>	<b>Python program to copy odd lines of one file to other.</b>	<b>57</b>	
<b>3</b>	<b>30/1/22</b>	<b>Write a Python program to read each row from a given csv file and print a list of strings.</b>	<b>59</b>	
<b>4</b>	<b>30/1/22</b>	<b>Write a Python program to read specific columns of a given CSV file.</b>	<b>63</b>	
<b>5</b>	<b>30/1/22</b>	<b>Write a Python program to write a Python dictionary to a csv file.</b>	<b>65</b>	

# I. COURSE OUTCOME 1(CO1)

## **PROGRAM NO:1**

**DATE:29/11/2021**

**AIM:**Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE .

A code editor is a tool that is used to write and edit code. They are usually lightweight and can be great for learning. However, once your program gets larger, you need to test and debug your code, that's where IDEs come in.

An IDE (Integrated Development Environment) understand your code much better than a text editor. It usually provides features such as build automation, code linting, testing and debugging. This can significantly speed up your work. The downside is that IDEs can be complicated to use.

Some IDE's are:

- Eclipse PyDev. ...
- IDLE. ...
- Wing. ...
- Emacs. ...
- Visual Studio Code. ...
- Sublime Text:
- Thonny
- PyCharm
- Atom
- Spyder

## IDLE

IDLE is Python's Integrated Development and Learning Environment.

IDLE has the following features:

- coded in 100% pure Python, using the `tkinter` GUI toolkit
- cross-platform: works mostly the same on Windows, Unix, and macOS
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features
- search within any window, replace within editor windows, and search through multiple files (grep)
- debugger with persistent breakpoints, stepping, and viewing of global and local namespaces
- configuration, browsers, and other dialogs

**PROGRAM NO:2****DATE:29/11/2021****AIM:**Display future leap years from current year to a final year entered by user.**PROGRAM**

```
y1=int(input("enter the year1 :"));
y2=int(input("enter the year2 :"));
print("Future leap years:")
for i in range(y1,y2):
    if i%4==0 and i%100!=0:
        print(i)
```

**OUTPUT:**

enter the year1 :2021

enter the year2 :2060

Future leap years:

2024

2028

2032

2036

2040

2044

2048

2052

2056



### **PROGRAM NO:3**

**DATE:29/11/2021**

**AIM:**List comprehensions:

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

### **PROGRAM**

```
list1=[]
l=[2,-2,45,65,-64,32,-111]

print("positive numbers are :")

for i in range(len(l)):
    if(l[i]>0):
        list1.append(l[i])
print(list1)
```

### **OUTPUT:**

positive numbers are :

[2, 45, 65, 32]

- **Square of N number**

### **PROGRAM**

```
n=int(input("enter the limit\n"))
s=[ i**2 for i in range(1,n+1)]
print(s)
```

### **OUTPUT**

enter the limit

7

[1, 4, 9, 16, 25, 36, 49]

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

### **PROGRAM**

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

### **OUTPUT**

Enter the word :hai World

The vowels in the word is: ['a'] ['i'] ['o']

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

### **PROGRAM**

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

### **OUTPUT**

Enter a word:Anandhu

Ordinal values for each elements:

A:65 n:110 a:97 n:110 d:100 h:104 u:117

**PROGRAM NO:4****DATE:29/11/2021****AIM:**Count the occurrences of each word in a line of text.**PROGRAM**

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

**OUTPUT**

Enter a string : This is python

count of the occurrence:[('This', 1), ('is', 1), ('python', 1)]

## **PROGRAM NO:5**

**DATE:29/11/2021**

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

### **PROGRAM**

```
lt=[]  
n1=int(input("Enter a limit:"))  
print("Enter values")  
for i in range(0,n1):  
    lt.append(int(input()))  
print("\nThe list is:\n")  
for i in range(0,len(lt)):  
    if lt[i]>=100:  
        print("over")  
    else:  
        print(lt[i])
```

### **OUTPUT**

Enter a limit:3

Enter values

23

456

1000

The list is:

23

over

over

**PROGRAM NO:6**

**DATE:29/11/2021**

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

**PROGRAM**

```
list1=['a','a','b','a']  
N=list1.count('a')  
print("occurance of a:",N)
```

**OUTPUT**

occurance of a: 3

**PROGRAM NO:7****DATE:29/11/2021**

**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**

```
l1=[1,3,5,7,9,11,34]
```

```
l2=[5,13,45,7,20,65,1]
```

```
s=int(0)
```

```
c=int(0)
```

```
if len(l1)==len(l2):
```

```
    print("same length")
```

```
else:
```

```
    print("different length")
```

```
for i in range(0,len(l1)):
```

```
    s=s+l1[i]
```

```
for i in range(0,len(l2)):
```

```
    c=c+l1[i]
```

```
if(s==c):
```

```
    print("equal sum")
```

```
else:
```

```
    print("not same sum")
```

```
print("Same elements are:")
```

```
l=[]
```

```
for i in range(0,len(l1)):
```

```
for j in range(0,len(l2)):
    if l1[i]==l2[j]:
        l.append(l1[i] and l2[j])
    else:
        continue
print(l)
```

## **OUTPUT**

different length

not same sum

Same elements are:

[5, 7]

**PROGRAM NO:8****DATE:29/11/2021**

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

**PROGRAM**

```
str="onion";  
str1=str[0];  
str2=str[1:];  
str3=str2.replace(str1,"$")  
print("Before replace:",str)  
print("After replace:")  
print(str1+str3)
```

**OUTPUT**

Before replace: onion

After replace:

oni\$n



**PROGRAM NO:9****DATE:29/11/2021**

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

**PROGRAM**

```
str=input("enter the string: ")
s1=str[0];
s2=str[-1];
print("after swap")
print(s2+str[1:-1]+s1)
```

**OUTPUT**

```
enter the string: javascript
after swap
tavascripj
```

**PROGRAM NO:10**

**DATE:29/11/2021**

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

**PROGRAM**

```
r=float(input("enter the radius of the circle: "))  
print("Area=",3.14*r*r);
```

**OUTPUT**

```
enter the radius of the circle: 2  
Area= 12.56
```

**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: "))
large=x
if (large<y) and (y>z):
    large = y
elif (large< z) and (y <z):
    large = z
print("The largest number is",large)
```

**OUTPUT**

```
Enter 1st number: 3
Enter 2nd number: 2
Enter 3rd number: 15
The largest number is 15
```

**PROGRAM NO:12**

**DATE:29/11/2021**

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

### **PROGRAM**

```
s1=input("enter the string with extension: ")
s2=s1.split(".")
print("extension is:")
print(s2[1])
```

### **OUTPUT**

```
enter the string with extension: first.py
extension is:
py
```

**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.

**PROGRAM**

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

**OUTPUT**

```
enter the color:red  
enter the color:blue  
enter the color:black  
first color: red  
last color: black
```

**PROGRAM NO:14**

**DATE:29/11/2021**

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

**PROGRAM**

```
n=input("enter n :");  
n1=n*2;  
n2=n*3;  
print("n+nn+nnn:")  
print(int(n)+int(n1)+int(n2));
```

**OUTPUT**

```
enter n :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.

**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**

```
{'Blue', 'White'}
```

**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**

```
str1=input("enter string 1: ")
str2=input("enter string 2: ")
s1=str1[0]
s2=str2[0]
print(s2+str1[1:]+ " "+s1+str2[1:])
```

**OUTPUT**

```
enter string 1: javascript
enter string 2: Python
Pavascript jython
```



**PROGRAM NO:17**

**DATE:29/12/2021**

**AIM:Sort dictionary in ascending and descending order.**

### **PROGRAM**

```
import operator
dic={"javascript":2,"java":1,"python":3}
print("Before Sorting",dic);
print("Sorting Ascending")
s1=sorted(dic.items(),key=operator.itemgetter(0))
print(s1)
print("Sorting descending")
s2=sorted(dic.items(),key=operator.itemgetter(0),reverse=True)
print(s2)
```

### **OUTPUT**

```
Before Sorting {'javascript': 2, 'java': 1, 'python': 3}
Sorting Ascending
[('java', 1), ('javascript', 2), ('python', 3)]
Sorting descending
[('python', 3), ('javascript', 2), ('java', 1)]
```

**PROGRAM NO:18**

**DATE:29/12/2021**

**AIM:**Merge two dictionaries.

**PROGRAM**

```
n1={"Anandhu":22,"Dist":"Ernakulam"}
n2={"Hari":23,"Dist":"Idukki"}
print("D1=",n1)
print("D2=",n2)
d1=n1.copy()
d1.update(n2)
print("After Merge ")
print(d1)
```

**OUTPUT**

```
D1= {'Anandhu': 22, 'Dist': 'Ernakulam'}
D2= {'Hari': 23, 'Dist': 'Idukki'}
After Merge
{'Anandhu': 22, 'Dist': 'Idukki', 'Hari': 23}
```

**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: 5
Enter 2nd number: 3
GCD : 1
```

**PROGRAM NO:20**

**DATE:29/11/2021**

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

**PROGRAM**

```
l=[2,27,54,33,45]
```

```
l2=[]
```

```
for i in l:
```

```
    if(i%2!=0):
```

```
        l2.append(int(i))
```

```
print("after removing even no.s")
```

```
print(l2)
```

**OUTPUT**

after removing even no.s

[27, 33, 45]

## II. COURSE OUTCOME 2(C02)

**PROGRAM NO:1**

**DATE:2/12/2021**

**AIM:**Program to find the factorial of a number.

**PROGRAM**

```
n=int(input("enter the number\n"))
```

```
f=1
```

```
print("Factorial of",n,"is")
```

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

```
    f=f*i
```

```
print(f)
```

**OUTPUT:**

enter the number

5

Factorial of 5 is

120

**PROGRAM NO:2**

**DATE:2/12/2021**

**AIM:**Generate Fibonacci series of N terms.

**PROGRAM**

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

```
f1=0
```

```
f2=1
```

```
f3=0
```

```
print("Fibonacci series:\n")
```

```
for i in range(n):
```

```
    print(f3)
```

```
    f1=f2
```

```
    f2=f3
```

```
    f3=f1+f2
```

**OUTPUT:**

enter the number :6

Fibonacci series:

0

1

1

2

3

5

**PROGRAM NO:3**

**DATE:2/12/2021**

**AIM:**Find the sum of all items in a list.

**PROGRAM**

```
l1=[2,3,5,15]
```

```
print("Sum of elements in the list is:\n")
```

```
print(sum(l1))
```

**OUTPUT:**

Sum of elements in the list is:

25

**PROGRAM NO:4****DATE:2/12/2021**

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

**PROGRAM**

```
from math import sqrt as s
print("Numbers are:")
for i in range(999,10000):
    if s(i)==int(s(i)) and i%2==0:
        print(i,end=" ")
```

**OUTPUT:**

Numbers are:

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:2/12/2021****AIM:**Display the given pyramid with step number accepted from user.**PROGRAM**

```
n=int(input("enter the limit\n"));  
print("The pattern is:");  
for i in range(n):  
    for j in range(1,i+2):  
        print(j,end="")  
    print()
```

**OUTPUT:**

enter the limit

5

The pattern is:

1

12

123

1234

12345

**PROGRAM NO:6**

**DATE:2/12/2021**

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

**PROGRAM**

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

```
fq= {}
```

```
for i in str1:
```

```
    if i in fq:
```

```
        fq[i] += 1
```

```
    else:
```

```
        fq[i] = 1
```

```
print ("Count of all characters : ",fq)
```

**OUTPUT**

Enter the string : Anandhu

Count of all characters : {'A': 1, 'n': 2, 'a': 1, 'd': 1, 'h': 1, 'u': 1}

**PROGRAM NO:7****DATE:2/12/2021**

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

**PROGRAM**

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

**OUTPUT:**

enter a string:Ride

new string is: Rideing

**PROGRAM NO:8****DATE:2/12/2021****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):  
    s=input("Enter element "+ str(x+1))  
    a.append(s)  
    m=len(a[0])  
    t=a[0]  
for i in a:  
    if(len(i)>m):  
        m=len(i)  
        t=i  
print("Longest Word is:",t)  
print("Length of longest word is:",m)
```

**OUTPUT:**

Enter the number of elements in list:3

Enter element1Anandhu

Enter element2Hari

Enter element3Athul

Longest Word is: Anandhu

Length of longest word is: 7

**PROGRAM NO:9**

**DATE:2/12/2021**

**AIM:**Construct following pattern using nested loop.

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

**PROGRAM**

```
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:5**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**PROGRAM NO:10**

**DATE:8/12/2021**

**AIM:**Generate all factors of a number. def print\_factors(x):

**PROGRAM**

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

**OUTPUT**

Enter the Number 20

Factors of 20 is

1  
2  
4  
5  
10  
20

**PROGRAM NO:11****DATE:8/12/2021****AIM:**Write lambda functions to find area of square, rectangle and triangle.**PROGRAM**

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

```
sq=lambda s:s*s
```

```
tri=lambda a,b:1/2*a*b
```

```
l,b=int(input("Enter the length and breadth of rectangle")),int(input())
```

```
print("Area of rectangle is :",rect(l,b))
```

```
s=int(input("Enter the side of square"))
```

```
print("Area of square is :",sq(s))
```

```
l,h=int(input("Enter the breadth and height of triangle")),int(input())
```

```
print("Area of triangle is :",tri(l,h))
```

**OUTPUT**

Enter the length and breadth of rectangle30

20

Area of rectangle is : 600

Enter the side of square15

Area of square is : 225

Enter the breadth and height of triangle10

20

Area of triangle is : 100.0



### III. COURSE OUTCOME 3(CO3)

**PROGRAM NO:1**

**DATE:20/12/2021**

**AIM:**Work with built-in packages.

#### **Math Module**

- **CO3 math.py**

#### **PROGRAM**

```
import math
print("Factorial:",math.factorial(5))
print ("GCD:",math.gcd(4,12))
print ("sqrt:",math.sqrt(25))
```

#### **OUTPUT:**

```
Factorial: 120
GCD: 4
sqrt: 5.0
```

#### **Time Module**

- **CO3 time.py**

#### **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: 1639961383.5407367
Current time: Mon Dec 20 06:19:43 2021
Time After 30 sec: Mon Dec 20 06:20:13 2021
Time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=6,
tm_min=19, tm_sec=43, tm_wday=0, tm_yday=354, tm_isdst=0)
Time-current year: 2021
Time:-current month 12
Time:-current day 20
Time:-current hour 6
Time:-current minute 19
Time:-current sec 43
Time:-current week day 0
Time:-current year day 354
```

### **Date Time Module**

- **CO3\_dtime.py**

### **PROGRAM**

```
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("Today:",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("date-time comb:",dt)
```

#### **OUTPUT:**

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

```
Today: 2021-12-20
Year: 2021
Month: 12
Day: 20
2021-12-20
2 days, 0:00:00
2021-12-22
date-time comb: 2021-12-20 22:56:20.000067
```

### **Calendar Module**

- **CO3 cal.py**

#### **PROGRAM**

```
import calendar
mm=int(input("Enter month:"))
yy=int(input("Enter year:"))
```

```
print(calendar.month(yy,mm))
```

### OUTPUT:

```
Enter month:2
Enter year:2022
February 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
```

## Statistics Module

- CO3 statistics.py

### PROGRAM

```
import statistics

print("MEAN:",statistics.mean([2,4,7,6,10]))

print("MEDIAN:",statistics.median([1, 3, 5, 7, 9, 11, 13]))

print("MODE:",statistics.mode([1, 1, -3, 3, 7, -9]))

print("VARIENCE:",statistics.variance([1, 3, 5, 7, 9, 11]))

print("HARMONIC MEAN:",statistics.harmonic_mean([40, 60, 80]))
```

### OUTPUT:

```
MEAN: 5.8
MEDIAN: 7
MODE: 1
VARIENCE: 14
HARMONIC MEAN: 55.38461538461538
```

## Random Module

- CO3\_random.py

## PROGRAM

```
import random

print("seed:",random.seed(10))
print("random float:",random.random())

mylist = ["apple", "banana", "cherry"]

print("sample:",random.sample(mylist, k=2))

print(random.random())

mylist2 = ["apple", "banana", "cherry"]
random.shuffle(mylist2)
print("shuffle:",mylist2)

mylist3 = ["apple", "banana", "cherry"]

print("choice:",random.choice(mylist3))
```

## OUTPUT:

```
seed: None
random float: 0.5714025946899135
sample: ['banana', 'cherry']
0.5780913011344704
shuffle: ['cherry', 'banana', 'apple']
choice: banana
```

**PROGRAM NO:2**

**DATE:20/12/2021**

**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)

- CO3 area.py

### **PROGRAM**

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

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

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

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

```
r=int(input("\nEnter the radius of sphere:"))
```

```
print("Area=",sphere.area(r))
print("perimeter=",sphere.perimeter(r))
```

### **graphics package:**

#### **circle.py**

#### **PROGRAM**

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

#### **rectangle.py**

#### **PROGRAM**

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

#### **cuboid.py**

#### **PROGRAM**

```
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))
```

#### **sphere.py**

## PROGRAM

```
def area(r):  
    return(4*3.14*r*r)  
def perimeter(r):  
    return(2*3.14*r)
```

## OUTPUT:

Enter the length of rectangle:4  
Enter the breadth of rectangle:3  
Area= 12  
Perimeter= 14

Enter the radius of circle:2  
Area= 12.56  
Perimeter= 12.56

Enter the length of cuboid:4  
Enter the width of cuboid:3  
Enter the height of cuboid:5  
Enter the breadth of cuboid:6  
Area= 94  
perimeter= 60

Enter the radius of sphere:4  
Area= 200.96  
perimeter= 25.12



## IV. COURSE OUTCOME 4(CO4)

**PROGRAM NO:1**

**DATE:11/1/2022**

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

**PROGRAM**

```
class rectangle:
    def __init__(self,le,br):
        self.le=le
        self.br=br
    def area(self):
        ar=self.le*self.br
        return(ar)

    def perimeter(self):
        pr=2*(self.le+self.br)
        return(pr)

print("First rectangle")
l1=int(input("Enter the length :"))
b1=int(input("Enter the breadth :"))
p1=rectangle(l1,b1)
print("Area=",p1.area())
print("Perimeter=",p1.perimeter())
print("-----")
print("Second rectangle")
l2=int(input("Enter the length :"))
b2=int(input("Enter the breadth :"))
p2=rectangle(l2,b2)
```

```
print("Area=",p2.area())
print("Perimeter=",p2.perimeter())
if(p1.area()>p2.area()):
    print("Rectangle 1 area is large")
else:
    print("Rectangle 2 area is large")
```

### **OUTPUT:**

First rectangle

Enter the length :2

Enter the breadth :4

Area= 8

Perimeter= 12

-----

Second rectangle

Enter the length :3

Enter the breadth :6

Area= 18

Perimeter= 18

Rectangle 2 area is large

**PROGRAM NO:2****DATE:11/1/2022**

**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,an,name,tp,bal=0):
        self.an=an
        self.name=name
        self.tp=tp
        self.bal=bal
    def deposit(self,bal):
        self.bal=self.bal+bal
    def withdraw(self,bal):
        if(bal<self.bal):
            self.bal=self.bal-bal
        else:
            print("Insufficient Balance")
    def dis(self):
        print("Name:",self.name)
        print("Account number:",self.an)
        print("Account Type:",self.tp)
        print("Balance Ammount:",self.bal)
```

```
an=int(input("Enter the account number: "))
name=input("Enter the name: ")
tp=input("Enter Account type:(AC/C)")
bn=bank(an,name,tp)
print("Enter your option: ")
print("1.Deposit")
print("2.Withdraw")
print("3.Display")
print("4.Exit")
while(True):
    opt=input("Enter your option: ")
    if(opt=='1'):
        print("deposit")
        am=int(input("Enter the amount to deposit: "))
        bn.deposit(am)
    elif(opt=='2'):
        print("withdraw")
        am1=int(input("Enter the amount to withdraw: "))
        bn.withdraw(am1)
    elif(opt=='3'):
        bn.dis()
    elif(opt=='4'):
        break
    else:
        print("Invalid Input")
```

## OUTPUT:

Enter the account number: 123

Enter the name: Anandhu

Enter Account type:(AC/C)AC

Enter your option:

1.Deposit

2.Withdraw

3.Display

4.Exit

Enter your option: 1

deposit

Enter the amount to deposit: 1000

Enter your option: 3

Name: Anandhu

Account number: 123

Account Type: AC

Balance Ammount: 1000

Enter your option: 2

withdraw

Enter the amount to withdraw: 500

Enter your option: 3

Name: Anandhu

Account number: 123

Account Type: AC

Balance Ammount: 500

Enter your option: 2

withdraw

Enter the amount to withdraw: 550

Insufficient Balance

Enter your option: 4

**PROGRAM NO:3****DATE:11/1/2022**

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

**PROGRAM**

```
class rect:
    def __init__(self,l,b):
        self.__l=l
        self.__b=b
    def __lt__(self,a):
        a1=self.__l*self.__b
        a2=a.__l*a.__b
        if(a1<a2):
            return(True)
print("recatngle1")
l1=int(input("Enter the length: "))
b1=int(input("Enter the breadth: "))
print("recatngle2")
l2=int(input("Enter the length: "))
b2=int(input("Enter the breadth: "))
ar1=rect(l1,b1)
ar2=rect(l2,b2)
if(ar1<ar2):
    print("rect1 area less")
else:
    print("rect2 area less")
```

**OUTPUT:**

recatngle1

Enter the length: 4

Enter the breadth: 6

recatngle2

Enter the length: 2

Enter the breadth: 5

rect2 area less



**PROGRAM NO:4****DATE:11/1/2022**

**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,hr,mt,sec):
        self.__hr=hr
        self.__mt=mt
        self.__sec=sec
    def __add__(self,a):
        h=self.__hr+a.__hr
        m=self.__mt+a.__mt
        if(m>60):
            q=int(m/60)
            r=m%60
            h=h+q
            m=r
        s=self.__sec+a.__sec
        if(s>60):
            q1=int(s/60)
            r1=s%60
            m=m+q1
            s=r1
        print("Added Time is:")
        print(h,":",m,":",s)
```

```
print("Time 1")
h1=int(input("Enter Hour: "))
m1=int(input("Enter Minute: "))
s1=int(input("Enter Second: "))
tm1=time(h1,m1,s1)
print("Time 2")
h2=int(input("Enter Hour: "))
m2=int(input("Enter Minute: "))
s2=int(input("Enter Second: "))
tm2=time(h2,m2,s2)
tm1+tm2
```

### **OUTPUT:**

```
Time 1
Enter Hour: 2
Enter Minute: 45
Enter Second: 33
Time 2
Enter Hour: 3
Enter Minute: 44
Enter Second: 22
Added Time is:
6 : 29 : 55
```

**PROGRAM NO:5****DATE:12/1/2022**

**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,name):
        self.name=name
    def display1(self):
        print("Publisher:",self.name)

class book(publisher):
    def TA(self,title,author):
        self.title=title
        self.author=author

    def display1(self):
        print("Title:",self.title)
        print("Author:",self.author)

class python(book):
    def __init__(self,price,no_of_page,name):
        self.price=price
        self.no_of_page=no_of_page
        super().__init__(name)
```

```
def display(self):  
    print("Price:",self.price)  
    print("No Of Pages:",self.no_of_page)
```

```
obj=python(1000,500,"Van Bossom")  
obj.TA("Python","Van Bossom")  
obj.display1()  
obj.display()
```

### **OUTPUT:**

Title: Python

Author: Van Bossom

Price: 1000

No Of Pages: 500

## V. COURSE OUTCOME 5(CO5)

**PROGRAM NO:1**

**DATE:19/1/2022**

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

**PROGRAM**

```
f1=open("secfile.txt","w")
f1.write("This is my first file in python.\n want to work with files \n This is my third
line")
f1=open("secfile.txt","r")
ff=f1.readlines()
print(ff)
```

**Output**

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

## PROGRAM NO:2

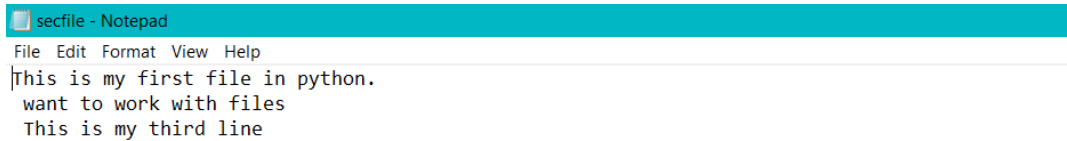
**DATE:19/1/2022**

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

## PROGRAM

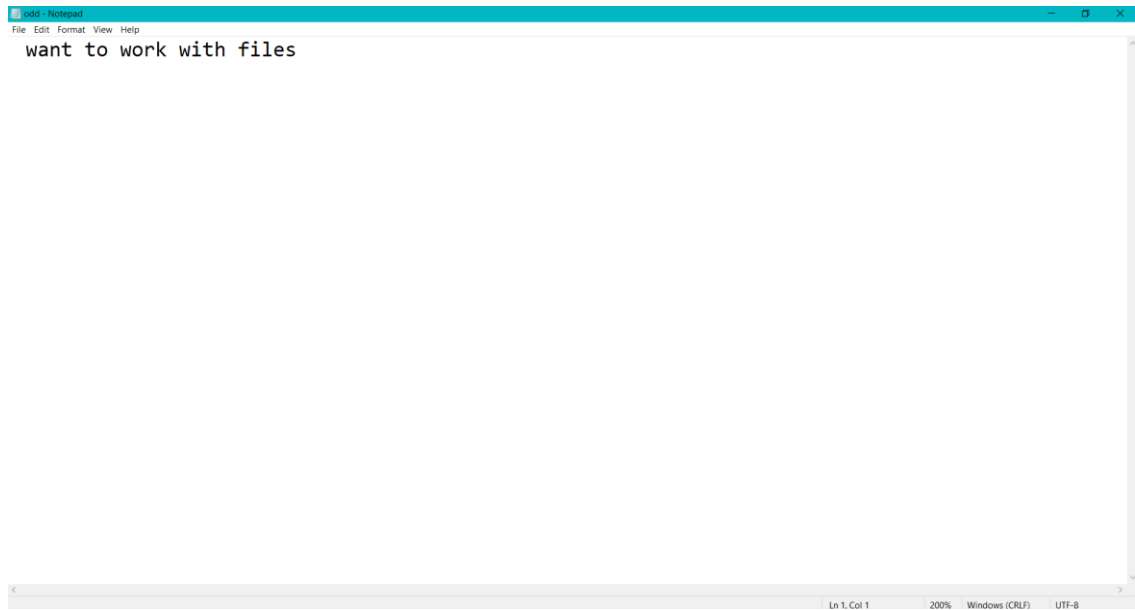
```
f1=open("secfile.txt","r")
ff=f1.readlines()
with open("odd.txt","w") as f2:
    for x in range(0,len(ff)):
        if(x%2!=0):
            f2.write(ff[x])
```

### secfile.txt



```
secfile - Notepad
File Edit Format View Help
This is my first file in python.
want to work with files
This is my third line
```

## Output:



### **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

# csv file name
filename = "username.csv"

# initializing the titles and rows list
fields = []
rows = []

# reading csv file
cf=open(filename, 'r')

    # creating a csv reader object
csvreader = csv.reader(cf)

    # extracting field names through first row
fields = next(cf)
print(fields)

print("\n.....\n")

# extracting each data row one by one
for r in csvreader:
    rows.append(r)
```



```

#print the list containing the rows of csv file
print(rows)
print("\n.....\n")
print('\nFirst 3 rows are:\n')
for r in rows[:3]:
    print(*r)
print("\n.....\n")
print("\nThe file content\n")
for sl in rows:
    for l in sl:
        print(l),
        #print(l,end=" ")
    print()

cf.close()

```

### username.csv

	A	B	C	D	E	F	G	H	I	J	K	L
1	Username; Identifier;Firstname;Lastname											
2	booker12;9012;Rachel;Booker											
3	grey07;2070;Laura;Grey											
4	johnson81;4081;Craig;Johnson											
5	jenkins46;9346;Mary;Jenkins											
6	smith79;5079;Jamie;Smith											
7												
8												
9												
10												
11												
12												
13												

### 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']]
```

.....

First 3 rows are:

```
booker12;9012;Rachel;Booker  
grey07;2070;Laura;Grey  
johnson81;4081;Craig;Johnson
```

.....

The file content

```
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 CSV file and print the content of the columns.

## PROGRAM

```
import csv

filename = "Names.csv"

cf=open(filename, 'r')

#csvreader = csv.reader(cf)

data = csv.DictReader(cf)
print("No Company")

for r in data:

    print(r['No'], r['Company'])
```

### Names.csv

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

**Output:**

No Company

1 Ferrari

2 Porsche

3 Bugatti

4 Rolls Royce

5 BMW

## **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.

## **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)

    #print(".....")
filename = "Names1.csv"

cf=open(filename, '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

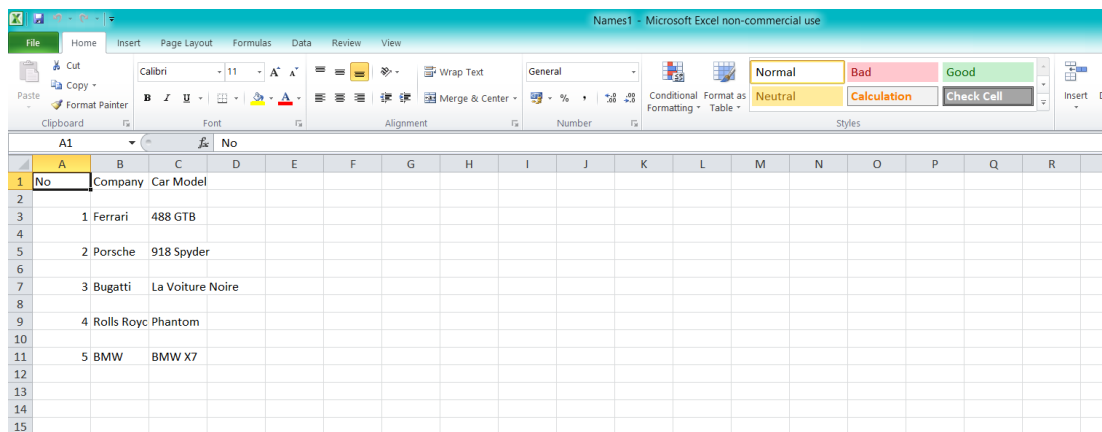
2 Porsche 918 Spyder

3 Bugatti La Voiture Noire

4 Rolls Royce Phantom

5 BMW BMW X7

## Names1.csv



The screenshot shows a Microsoft Excel spreadsheet titled "Names1 - Microsoft Excel non-commercial use". The spreadsheet contains a table with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	No	Company	Car Model															
2																		
3		1	Ferrari	488 GTB														
4																		
5		2	Porsche	918 Spyder														
6																		
7		3	Bugatti	La Voiture Noire														
8																		
9		4	Rolls Royce	Phantom														
10																		
11		5	BMW	BMW X7														
12																		
13																		
14																		
15																		