SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-23

MCA PROGRAMMING LABORATORY RECORD

Submitted by

FABIN FRANCIS

REG NO: SNG21MCA-2013

in partial fulfillment for the award of the degree in

MASTER OF COMPUTER APPLICATION

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

MCA PROGRAMMING LABORATORY RECORD

Certified that this is a Bonafide record of practical work done by ***Fabin Francis*** *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-2013 on ------------

External Examiner InternalExaminer

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 a nd 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.

<u>Top Python IDE’s</u>

- PyCharm
- Spyder
- Eclipse PyDev
- Wing
- IDLE

<u>PyCharm</u>

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.

```
print("Future Leap years")
print("enter the starting year")
start=int(input())
print("enter the final year")
end=int(input())
print("start year",start,"\nend year",end)
for y in range(start,end):
    if (y % 4 == 0) and (y % 100 != 0) or (y % 400 == 0):
     print(y)
```

OUTPUT

```
Future Leap years
enter the starting year
1692
enter the final year
1710
start year 1692
end year 1710
1692
1696
1704
1708
```

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

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

```
 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 N number**

```
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:4
Result: [1, 4, 9, 16]
```

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

```
list1=[]
w1=input("Enter the word:")
for i in w1:
  if i in 'aeiouAEIOU':
      list1.append(i)
 print("Resultant list:",list1)
```

OUTPUT

```
Enter the word:hello world
 Resultant list: ['e', 'o', 'o']
```

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

```
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

```
str1=input("Enter the text:")
list1=str1.split(" ")
print(list1)
for i in list1:
 count=list1.count(i)
 print(" ",i," ",count)
```

OUTPUT

```
 Enter the text:how is is
['how', 'is', 'is']
  how   1
  is   2
  is   2
```

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

```
list1=[]
n1=int(input("Enter the limit:"))
for i in range(n1):
 n2=int(input("Enter the number:"))


 if n2>100:
  list1.append("over")
 else:
  list1.append(n2)
  print(list1)
```

OUTPUT

```
Enter the limit:3
Enter the number:300
Enter the number:3
Enter the number:5
['over', 3, 5]
```

PROGRAM NO: 6

DATE:24/11/2021

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

```
 list1=["a","b","c","a","a"]
print("List:\n",list1)
print("Occurence of a:",list1.count('a'))
```

OUTPUT

```
  List:
 ['a', 'b', 'c', 'a', 'a']
Occurence of a: 3
```

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

```
list1=[]

list2=[]

s1=0

s2=0

n1=int(input("Enter the number of elements in list1:"))

for i in range(n1):

    x=int(input("Enter the elements:"))

  s1=s1+x

    list1.append(x)

n2=int(input("\nEnter the number of elements in list2:"))

for i in range(n2):

    x=int(input("Enter the elements:"))

    s2=s2+x

    list2.append(x)

print("\nlist1\n",list1)
print("\nlist2\n",list2)



if len(list1)==len(list2):
  print("\nLength of 2 list is same")
else:
  print("\nLength of 2 list is not same")
if s1 == s2:
 print("Sum is equal")
else:
print("Sum is not equal")
print("common elements:",set(list1).intersection(set(list2)))
```

OUTPUT

```
Enter the number of elements in list1:3
Enter the elements:3
Enter the elements:5
Enter the elements:7
Enter the number of elements in list2:3
Enter the elements:6
Enter the elements:7
Enter the elements:3
list1
 [3, 5, 7]
list2
 [6, 7, 3]
Length of 2 list is same
Sum is not equal
common elements: {3, 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]

```
string1=input("Enter the string:")
print("Old string:",string1)
first=string1[0]
print("New String:",first+string1[1:].replace(first,'$'))
```

OUTPUT

```
Enter the string:hai hello
Old string: hai hello
New String: hai $ello
```

PROGRAM NO: 9

DATE:24/11/2021

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

```
string1=input("Enter a string:")
print("Old string:",string1)
first=string1[0]

last=string1[-1]
print("New string:",last+string1[1:-1]+first)
```

OUTPUT

```
Enter a string:hello world
Old string: hello world
New string: dello worlh
```

PROGRAM NO: 10

DATE:24/11/2021

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

```
 r=float(input("Enter the radius:"))
print("radius=",r)
area=3.14*r*r
print("Area=",'%.2f'%area)
```

OUTPUT

```
Enter the radius:2
radius= 2.0
Area= 12.56
```

PROGRAM NO: 11

DATE:29/11/2021

AIM: Find biggest of 3 numbers entered

```
n1=int(input("Enter the first number:"))
n2=int(input("Enter the second number:"))

n3=int(input("Enter the third number:"))
large=n1
if(n2>large):
 large=n2
if(n3>large):
 large=n3
print("Largest number is:",large)
```

OUTPUT

```
Enter the first number:3
Enter the second number:44
Enter the third number:22
```

```
Largest number is: 44
```

PROGRAM NO: 12

DATE:29/11/2021

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

```
file1=input("Enter the file name:")
ext=file1.split('.')
print("File Name=",file1)
print("File Extension=",ext[-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.

```
list1=[]

n=int(input("Enter the number of colors:"))

for i in range(n):

 x=input("Enter the color:")

 list1.append(x)

print("\nList:",list1)

print("First color:",list1[0])

print("last color:",list1[n-1])
```

OUTPUT

```
Enter the number of colors:3
Enter the color:blue
Enter the color:black

Enter the color:yellow
List: ['blue', 'black', 'yellow']
First color: blue
last color: yellow
```

PROGRAM NO: 14

DATE:29/11/2021

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

```
n=input("Enter the number:")
print("Number is",n)
print("Result=",int(n)+ int(n*2) +  int(n*3))
```

OUTPUT

```
Enter the number:4
Number is 4
Result= 492
```

PROGRAM NO: 15

DATE:29/11/2021

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

```
list1=["black","green","red"]
list2=["black","blue","red"]
print("Difference:",set(list1) ^ set(list2))
```

OUTPUT

```
Difference: {'green', '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.

```
str1=input("Enter the first string:")
str2=input("Enter the second string:")
c1=str1[0]
c2=str2[0]
print("\nString 1:",str1)
print("string 2:",str2)
print("Resultant string:",c2+str1[1:]+" "+c1+str2[1:])
```

OUTPUT

```
Enter the first string:hello
Enter the second string:world

String 1: hello
string 2: world
Resultant string: wello horld
```

PROGRAM NO: 17

DATE:29/11/2021

AIM: Sort dictionary in ascending and descending order.

```
import operator
d={5:22,7:4,3:5}
print("Old Dictionary:\n",d)

print("\nSorted dictionary by value in ascending order:")
print(sorted(d.items(),key=operator.itemgetter(1)))

print("\nSorted dictionary by value in descending order:")
rev=dict(sorted(d.items(),key=operator.itemgetter(1),reverse=True))
print(rev)
```

OUTPUT

```
Old Dictionary:
 {5: 22, 7: 4, 3: 5}

Sorted dictionary by value in ascending order:
[(7, 4), (3, 5), (5, 22)]

Sorted dictionary by value in descending order:
{5: 22, 3: 5, 7: 4}
```

PROGRAM NO: 18

DATE:29/11/2021

AIM: Merge two dictionaries

```
d1={1:4,2:5,3:8}
print("First dictionary:",d1)
d2={1:6,2:8,4:9}
```

```
print("Second dictionary:",d2)
d3=d1.copy()
d3.update(d2)
print("Merged dictionary:",d3)
```

OUTPUT

```
First dictionary: {1: 4, 2: 5, 3: 8}
Second dictionary: {1: 6, 2: 8, 4: 9}
Merged dictionary: {1: 6, 2: 8, 3: 8, 4: 9}
```

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:20
Enter the second number:40

Gcd= 20
```

PROGRAM NO:20

DATE:29/11/2021

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

```
list1=[]
list2=[]
n=int(input("Enter the limit:"))
for i in range(n):
 x=int(input("Enter the element:"))
 list1.append(x)
 if x%2!=0:
  list2.append(x)
print("\nEntered List:",list1)
print("\nResultant List:",list2)
```

OUTPUT

```
Enter the limit:4
Enter the element:11

Enter the element:22
Enter the element:33
Enter the element:44

Entered List: [11, 22, 33, 44]
Resultant List: [11, 33]
```

II. COURSE OUTCOME 2(CO2)

PROGRAM NO: 1

DATE:1/12/2021

AIM: Program to find the factorial of a number

```
n1=int(input("Enter the number:"))
f=1
for i in range(1,n1+1):
    f=f*i
print("Factorial of",n1,"is:",f)
```

OUTPUT

```
Enter the number:6
Factorial of 6 is: 720
```

PROGRAM NO: 2

DATE:1/12/2021

AIM: Generate Fibonacci series of N terms

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

f=0

s=1

r=0

c=1


print("Fibonacci series:")

while c<=n:

   print(r,end=" ")

   c=c+1

   f=s

   s=r

   r=f+s
```

OUTPUT

```
Enter the limit:4
Fibonacci series:
0 1 1 2
```

PROGRAM NO: 3

DATE:1/12/2021

AIM: Find the sum of all items in a list

```
list1=[]
s=0
n=int(input("Enter the limit:"))
for i in range(n):
   x=int(input("Enter the number:"))
   list1.append(x)
   s=s+x
print("List:",list1)

print("Sum=",s)
```

OUTPUT

```
Enter the limit:4
Enter the number:2
Enter the number:3
Enter the number:2
Enter the number:1
List: [2, 3, 2, 1]
Sum= 8
```

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.

```
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**.**

```
 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’ 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

```
   list1=[]
n=int(input("Enter the number of strings:"))
for i in range(n):
 str=input("Enter the string:")
 list1.append(str);
lword=list1[0]
max=len(list1[0])
for i in list1:
 if(len(i)>max):
   max=len(i)
   lword=i
print("Longest word:",lword)

print("Length:",max)
```

OUTPUT

```
Enter the number of strings:3
Enter the string:hai
Enter the string:hello
Enter the string:hi
Longest word: hello
Length: 5
```

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

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(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.

```
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("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
```

<u>Math module</u>

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

OUTPUT

```
24
3
3.0
```

<u>Calendar module</u>

```
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
```

<u>DateTime module</u>

```
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
print(statistics.mean([3,4,3]))
print(statistics.median([1, 3, 5, 7, 9, 11, 13]))
print(statistics.mode([1, 1, -3, 3, 7, -9]))
print(statistics.variance([1, 3, 5, 7, 9, 11]))


print(statistics.stdev([1, 3, 5, 7, 9, 11]))
```

OUTPUT

```
3.3333333333333335
7
1
14
3.7416573867739413
```

Random module

```
import random
random.seed(10)
print(random.random())
mylist = ["apple", "banana", "cherry"]
print(random.sample(mylist, k=2))
print(random.random())
mylist2 = ["apple", "banana", "cherry"]
random.shuffle(mylist2)
print(mylist2)
mylist3 = ["apple", "banana", "cherry"]
print(random.choice(mylist3))
```

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 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,length,breadth):
    self.length=length
    self.breadth=breadth

  def area(self):
    area=self.length*self.breadth
    print("Area=",area)
    return(area)

  def perimeter(self):
    per=2*(self.length+self.breadth)
    print("Perimeter=",per)

print("First Rectangle:")
b1=rectangle(2,2)
a1=b1.area()
b1.perimeter()

print("\nSecond Rectangle:")
b2=rectangle(3,3)
a2=b2.area()
b2.perimeter()

if a1 > a2:
 print("\nArea of first rectangle is larger")
else:
 print("\nArea of second rectangle is larger")
```

OUTPUT

```
First Rectangle:
Area= 4
Perimeter= 8

Second Rectangle:
```

```
Area= 9
Perimeter= 12

Area of second rectangle is larger
```

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:1233
Enter the name:Alan
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:200

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:1

Account Info:
Account Number: 1233
Account Name: Alan
Account Type: savings
Account Balance: 2200

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:3
Enter the amount to withdraw:100
RS- 100 Withdrawn successfully

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:1

Account Info:
Account Number: 1233
Account Name: Alan
Account Type: savings
Account Balance: 2100

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:2
Enter the breadth:3
Area= 6

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,second):
   print("\nHour:",self.__hour + second.__hour)

   if self.__minute + second.__minute > 60:
    h1=(self.__minute + second.__minute)//60
    m1=(self.__minute + second.__minute)%60
    print("Minutes:",h1," hour ",m1," minutes")
   else:
    print("Minutes:",self.__minute + second.__minute)

   if self.__second+second.__second > 60:
    m1=(self.__second+second.__second)//60
    s1=(self.__second+second.__second)%60
    print("seconds:",m1," minutes ",s1," seconds")
   else:
    print("Seconds:",self.__second + second.__second)
```

```
hour1=int(input("Enter the hour:"))
minute1=int(input("Enter the minutes:"))
sec1=int(input("Enter the second:"))

obj1=time(hour1,minute1,sec1)

hour2=int(input("\nEnter the hour:"))
minute2=int(input("Enter the minutes:"))
sec2=int(input("Enter the second:"))

obj2=time(hour2,minute2,sec2)

obj1 + obj2
```

OUTPUT

```
Enter the hour:3
Enter the minutes:44
Enter the second:35

Enter the hour:4
Enter the minutes:55
Enter the second:45

Hour: 7
Minutes: 1  hour  39  minutes
seconds: 1  minutes  20  seconds
```

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: Flames
Author Name: K D
Price: 450
No of pages: 72
```

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
```