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 LABORATORY RECORD

Submitted by

ABHIRAM KS

REG NO: SNG21MCA-2001

in partial fulfillment for the award of the degree in

MASTEROF COMPUTERAPPLICATIONS

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

20 MCA 132 PROGRAMMING LABORATORY RECORD

Certified that this is a Bonafide record of practical work done by ***ABHIRAM KS*** *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 2021-22.*

Kadayiruppu

Date: ………………. Course Instructor

Head of the Department

Prof.Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2001 on ------------

External Examiner Internal Examiner

I. COURSE OUTCOME 1(CO1)

PROGRAM NO: 1

DATE:24/11/2021

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

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

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

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

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

<u>Top Python IDE’s</u>

- PyCharm
- Spyder
- Eclipse PyDev
- Wing
- IDLE

IDLE

is Python’s Integrated Developmentand

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:24/11/2021

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

```
 a=int(input("enter the starting year"))
b=int(input("enter tne last year"))
if(a<b):
print("leap year is",end="")
fori in range(a,b):
if(i%4==0 and i%100!=0):
print(i,end="")
  print()
```

OUTPUT

```
enter the starting year2001

 entertne last year2030

 leap year is2004

 2008
 2012
 2016
 2020
 2024
 2028
```

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

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

```
list=[10,-9,-25,66,3,7]
r=[n for n in list if n>=0]
print(r)
```

OUTPUT

```
[10, 66, 3, 7]
```

- **Square of Nnumber**

```
n=int(input("enter the number"))
square=[i**2 for i in range(1,n+1)]
print("squreis",square)
```

OUTPUT

```
enter the number6
squre is [1, 4, 9, 16, 25, 36]
```

- **Form a list of vowels selected from a givenword**

```
n=input("enter the word")
print("the strings are",n)
print("the vowels are",end="")
fori in n:
ifi in 'aeiouAEIOU':
   print([i],end="")
```

OUTPUT

```
enter the word javascript
the strings are    javascript
the vowels are['a']['a']['i']
```

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

```
w=input("Enter a word:")

print("Ordinal values corresponding to each element is:")

fori in w:

print(i,end=":")

print(ord(i),end=" ")
```

OUTPUT

```
Enter a word:javascript

Ordinal values corresponding to each element is
:
j:106 a:97 v:118 a:97 s:115 c:99 r:114 i:105 p:112 t:116
```

PROGRAM NO: 4

DATE:24/11/2021

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

```
str1 = input("Enter a string : ")

wlist = str1.split()

count= []

fori in wlist:

count.append(wlist.count(i))

print("count of the occurrence:" + str(list(zip(wlist, count))))
```

OUTPUT

```
Enter a string :   java is power full

count of the occurrence:[('java', 1), ('is', 1), ('power', 1), ('full', 1)]
```

PROGRAM NO: 5

DATE:24/11/2021

AIM: Prompttheuserforalistofintegers.Forallvaluesgreaterthan100,store‘over’instead

```
n=[]

s=int(input("Enter a limit:"))

print("Enter   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   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

```
list=["a","a","b","a","a","b","a"]

str=list.count("a")

print(str)

print("count of occurrences of a :",str)
```

OUTPUT

```
5
count of occurrences of a : 5
```

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

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

b=[5,13,45,7,20,65,1]

s=int(0)

c=int(0)

iflen(a)==len(b):

print("Lists are of same length")

else:

print("Lists have different length")

fori in range(0,len(a) and len(b)):

s=s+a[i]

c=c+b[i]

  if(s==c):

  print("equal sum")

  else:

  print("not same sum")

  print("Elements that matched are:")

  l=[]

  fori in range(0,len(a)):

  for j in range(0,len(b)):
```

```
if a[i]==b[j]:

l.append(a[i] and b[j])

else:

continue

print(l)
```

OUTPUT

```
Lists are of same length

not same sum

Elements that matched are:

[1, 5, 7]
```

PROGRAM NO: 8

DATE:24/11/2021

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

```
str1="malayalam"

char=str1[0]

str1=str1.replace(char,'$')

str1=char+str1[1:]

  print(str1)
```

OUTPUT

```
  malayala$
```

PROGRAM NO: 9

DATE:24/11/2021

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

```
str = input("Enter a string:")

new_str = str[-1:] +str[1:-1] + str[:1]

print("New string : ",new_str)
```

OUTPUT

```
Enter a string:webprogram

New string :mebprograw
```

PROGRAM NO: 10

DATE:24/11/2021

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

```
pi=3.14
r=float(input("enter the number"))

radius=pi*r*r

print(radius)
```

OUTPUT

```
enter the number10

314.0
```

PROGRAM NO: 11

DATE:29/11/2021

AIM: Find biggest of 3 numbers entered

```
a=int(input("enter the first number"))

b=int(input("enter the 2nd number"))

c=int(input("enter the 3rd number"))

if(a>b and a>c):

larg=a

elif(b>a and b>c):

larg=b

else:

larg=c

print("the largest number is",larg)
```

OUTPUT

```
enter the first number30

enter the 2nd number35

enter the 3rd number33

the largest number is 35
```

PROGRAM NO: 12

DATE:29/11/2021

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

```
file=input("enter file name:")

f=file.split(".")

print("Extension of the file is:"+ f[-1])
```

OUTPUT

```
enter file name:large.py

Extension of the file 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 lastcolors.

```
a=[]

for i in range(3):

  b=input("enter the color")

a.append(b)

print(a)

print("first",a[0])

print("last",a[2])
```

OUTPUT

```
enter the color green
[' green']
enter the color red
[' green', ' red']
enter the colorwhite
[' green', ' red', 'white']
first   green
last white
```

PROGRAM NO: 14

DATE:29/11/2021

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

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

x=int("%s"%n)

y=int("%s%s"%(n,n))

z=int("%s%s%s"%(n,n,n))

print("n+nn+nnn:",x+y+z)
```

OUTPUT

```
enter the number:6
n+nn+nnn: 738
```

PROGRAM NO: 15

DATE:29/11/2021

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

```
color_list_1 = set(["White", "pink", "Red","Blue"])

color_list_2 = set(["Red", "Green","pink"])

print(color_list_1.difference(color_list_2))
```

OUTPUT

```
{'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.

```
a="python"

b="cpp"

p1=a[0]

p2=b[0]

c=b[0]+a[1:len(a)]+" "+a[0]+b[1:len(b)]

print(c)
```

OUTPUT

```
cythonppp
```

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.

```
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: 120

Enter 2nd number: 5

GCD : 5
```

PROGRAM NO:20

DATE: 29/11/2021

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

```
num = [6,8, 124, 125, 44, 18, 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: [6, 8, 124, 125, 44, 18, 27]

list after removing Even numbers: [125, 27]
```

II. COURSE OUTCOME 2(CO2)

PROGRAM NO: 1

DATE: 1/12/2021

AIM: Program to find the factorial of a number

```
n=int(input("enter the number"))
f=1
for i in range(1,n+1):
    f=f*i
  print("factor is",n,"=",f)
```

OUTPUT

```
enter the number5

factor is 5 = 120
```

PROGRAM NO: 2

DATE: 1/12/2021

AIM: Generate Fibonacci series of N terms

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

OUTPUT

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

PROGRAM NO: 3

DATE: 1/12/2021

AIM: Find the sum of all items in a list

```
list=[10,40,50,30]
total=sum(list)
print("total is",total)
```

OUTPUT

```
total is 130
```

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.

```
r=int(input("enter the number"))
for i in range(1,r+1):
for j in range(1,i+1):
print(i*j,end="")
  print()
```

OUTPUT

```
enter the number5
1
24
369
481216
510152025
```

PROGRAM NO: 6

DATE: 1/12/2021

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

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

freq = {}

for i in test_str:

ifi in freq:

freq[i] += 1

else:

freq[i] = 1

print ("Count of all characters : "+ str(freq))
```

OUTPUT

```
Enter the string :malayalam

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

PROGRAM NO: 7

DATE: 8/12/2021

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

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

OUTPUT

```
enter a string:game
inputed string is: game
theformated string is: gameing
```

PROGRAM NO: 8

DATE: 8/12/2021

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

```
a=[]
n= int(input("Enter the 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:1
Enter element 1java is power full
Longest Word: java is power full
Length of longest word : 18
```

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
```

Math module

```
import math
print("The value of pi ",math.pi)
import math as m
print("The value of pi ",m.pi)
from math import pi,sqrt
print(math.pi)
print(math.sqrt(5))
print(math.tan(30))
print(math.cos(45))
```

OUTPUT

```
The value of pi    3.141592653589793
The value of pi    3.141592653589793
3.141592653589793
2.23606797749979
-6.405331196646276
0.5253219888177297
```

Calendar module

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

OUTPUT

```
Enter month:1
Enter year:2022

    January 2022
```

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

DateTime module

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

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

OUTPUT

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

```
Microsecond: 67


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

Statistics module

```
import statistics
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())
print(random.uniform(20, 60))
lst = ["orange", "apple", "graphes"]
print(random.sample(lst, k=2))
print(random.random())
lst2 = ["orange", "apple", "graphes"]
```

```
random.shuffle(lst2)
print(lst2)
lst3 = ["orange", "apple", "graphes"]
print(random.choice(lst3))
```

OUTPUT

```
0.5714025946899135

37.155562187004584

['graphes', 'orange']

0.20609823213950174

['orange', 'graphes', 'apple']
```

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

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

rectangle module

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

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

- **3dgraphics package**

sphere module

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

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

cuboid module

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

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

Area

```
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

```
Enter length of cuboid:5
Enter width of cuboid:4
Enter height of cuboid:3
Enter breadth of cuboid:7
Area of cuboid= 94
perimeter of cuboid= 60
Enter the radius of sphere:5
Area of sphere= 314.0
perimeter of sphere= 31.400000000000002
Enter length of rectangle:3
Enter breadth of rectangle:4
Area of rectangle= 12
Perimeter of rectangle= 14
Enter radius of circle:6
Area of Circle: 113.03999999999999
Perimeter of Circle: 37.68
```

IV. COURSE OUTCOME 4(CO4)

PROGRAM NO: 1

DATE: 9/1/2022

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

```
class rectangle:
    def __init__(self,l,b):

        self.l=l

        self.b=b

    def area(self):
        area=self.l*self.b

        print("area of rectangle",area)

        return(area)

    def perimeter(self):

        per=2*(self.l+self.b)

        print("perimeter of rectangle",per)

        return(per)

r1=rectangle(7,8)

r2=rectangle(8,7)

a=r1.area()

r1.perimeter()

b=r2.area()

r2.perimeter()

if(a>b):
    print("Rectangle one area is greater",a)
else:
     print("Rectangle two area is greater",b)
```

OUTPUT

```
area of rectangle 56
perimeter of rectangle 30
area of rectangle 56
```

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 amount deposit:"))
     self.bal=self.bal+dep

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

acc_no=int(input("Enter Account Number:"))
acc_name=input("Enter name:")
acc_type=input("Enter account type(savings/current):")
balance=int(input("Enter 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("Exit")
break
else:
  print("Invalid Option")
```

OUTPUT

```
Enter Account Number:4567
Enter name:abhi
Enter account type(savings/current):savings
Enter initial balance:400
1.Account info
2.Deposit
3.Withdraw
4.Exit
Select your option:1

Account info:
Account number: 4567
Account name: abhi
Account type: savings
Account balance: 400

1.Account info
2.Deposit
3.Withdraw
4.Exit
Select your option:2
Enter amount deposit:5000
```

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

Account info:
Account number: 4567
Account name: abhi
Account type: savings
Account balance: 5400

1.Account info
2.Deposit
3.Withdraw
4.Exit
Select your option:3
Enter amount withdraw:3000
Rs 3000 Successfully Withdrawn

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

Account info:
Account number: 4567
Account name: abhi
Account type: savings
Account balance: 2400

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

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:5
Enter the breadth:6
Area= 30

Area of second 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,tm):
        print("Hour",self.__hour+tm.__hour)
        print("Minute",self.__minute+tm.__minute)
        print("Second",self.__second+tm.__second)
hr=int(input("Enter the hour"))
mins=int(input("Enter the min"))
sec=int(input("Enter the seco"))
obj=time(hr,mins,sec)
hr1=int(input("Enter the hour"))
mins1=int(input("Enter the min"))
sec1=int(input("Enter the seco"))
obj1=time(hr1,mins1,sec1)
obj+obj1
```

OUTPUT

```
Enter the hour3

Enter the min45

Enter the seco23

Enter the hour6

Enter the min45

Enter the seco54

Hour 9

Minute 90

Second 77
```

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,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("java script","Brendan Eich",1000,120)
p.display()
```

OUTPUT

```
Title: java script
Author: Brendan Eich
Price: 1000
No. of Pages: 120
```

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("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']

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("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])
```

OUTPUT

```
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.

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

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))
```

<u>cars.csv</u>

```
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
```