SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING

KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

ACADEMIC YEAR 2021-2022

MCA PROGRAMMING LABORATORY RECORD

Submitted by

DEVU K ANIL

REG NO: SNG21MCA-2012

in partial fulfillment for the award of the degree in

MASTER OF COMPUTER APPLICATIONS

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING KADAYIRUPPU, KOLENCHERY 682 311

(Affiliated to APJ Abdul Kalam Technological University)

MCA PROGRAMMING LABORATORY RECORD

Certified that this is a Bonafide record of practical work done by ***Devu K Anil*** *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 2021-2022.*

Kadayiruppu   Course Instructor

Date: ……………….

Head of the Department

Prof.Dr. SANDHYA R

Submitted for University Practical Examination

Reg No: SNG21MCA-2012 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.

Top Python IDE's

- PyCharm
- Spyder
- Eclipse PyDev
- Wing
- IDLE

PyCharm

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

- It is considered as an intelligent code editor, fast and safe refactoring, and smart code.

- Features for debugging, profiling, remote development, testing the code, auto code completion, quick fixing, error detection and tools of the database.

- Support for Popular web technologies, web frameworks, scientific libraries and version control.

PROGRAM NO: 2

DATE:24/11/2021

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

PROGRAM s

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

OUTPUT

```
enter start year:2021

enter end year:2050

leap year is 2024 2032 2036 2040 2044 2048
```

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

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

PROGRAM

```
 list1=[-10,20,35,-67,70]
list2=[]
for i in list1:
if i>0:
list2.append(i)
print("Resultant list",list2)
```

OUTPUT

```
Resultant list [20,35,70]
```

- **Square of N number**

PROGRAM

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

OUTPUT

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

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

PROGRAM

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

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

OUTPUT

```
Enter the string:python programming
 the actual string is python programming
 vowels are: o o a i
```

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

PROGRAM

```
w1=input("Enter the word:")
print("ordinal values corresponding to each element is:"
for i in w1:
 print(i,end ".")
 print(ord(i),end="")
```

OUTPUT

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

PROGRAM NO: 4

DATE:24/11/2021

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

PROGRAM

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

wordlist=str1.split()

count=[]

for w in wordlist:

count.append(wordlist.count(w))

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

OUTPUT

```
enter the string: python
count of the occurence: [('python', 1)]
```

PROGRAM NO: 5

DATE:24/11/2021

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

PROGRAM

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

The list after assinging:

24
over
```

PROGRAM NO: 6

DATE:24/11/2021

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

PROGRAM

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

OUTPUT

```
Count of  occurrence of  a: 2
```

PROGRAM NO: 7

DATE:24/11/2021

AIM: Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both

PROGRAM

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

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

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

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

OUTPUT

```
Lists are of same length
not same sum

Elements that matched are:
[1, 5, 7]
```

PROGRAM NO: 8

DATE:24/11/2021

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

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

OUTPUT

```
malayala$
```

PROGRAM NO: 9

DATE:24/11/2021

AIM: Create a string from given string where first and last characters exchanged.

PROGRAM

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

OUTPUT

```
enter a string:PYTHON
new string: NYTHOP
```

PROGRAM NO: 10

DATE:24/11/2021

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

PROGRAM

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

OUTPUT

```
input the radius:6
the area of the circle with radius is: 113.04
```

PROGRAM NO: 11

DATE:29/11/2021

AIM: Find biggest of 3 numbers entered

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

OUTPUT

```
Enter 1st number: 56

Enter 2nd number: 34

Enter 3rd number: 78

The largest number is 78
```

PROGRAM NO: 12

DATE:29/11/2021

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

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

OUTPUT

```
Enter  file name:hello.java

Extension of the file: 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.

PROGRAM

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

OUTPUT

```
enter the color:red
enter the color:green
enter the color:blue
['red', 'blue', 'green']
red
green
```

PROGRAM NO: 14

DATE:29/11/2021

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

```
n=int(input("enter a number:"))
x=int("%s"%n)

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

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

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

OUTPUT

```
enter a number:5

n+nn+nnn: 615
```

PROGRAM NO: 15

DATE:29/11/2021

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

```
color_list_1=set(["white","pink","red","blue"])

color_list_2=set(["red","green","pink"])

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

OUTPUT

```
{'white', 'blue'}
```

PROGRAM NO: 16

DATE:29/11/2021

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

```
a="python"

b="java"

p1=a[0]

p2=b[0]

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

print(c)
```

OUTPUT

```
jythonpava
```

PROGRAM NO: 17

DATE:29/11/2021

AIM: Sort dictionary in ascending and descending order.

PROGRAM

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

OUTPUT

```
Original dictionary :  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

Dictionary in ascending order by value  [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]

Dictionary in descending order by value :  {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
```

PROGRAM NO: 18

DATE:29/11/2021

AIM: Merge two dictionaries

PROGRAM

```
d1={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

```
Dictionary 1= {'a': 50, 'b': 150}

Dictionary 2= {'x': 250, 'y': 200}

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

PROGRAM NO: 19

DATE:29/11/2021

AIM: Find gcd of 2 numbers.

```
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 = [7,8, 120, 25, 44, 20, 27]
print( "Original list:",num)
num = [x for x in num if x%2!=0]
print("list after removing Even numbers:",num)
```

OUTPUT

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

list after removing Even numbers: [7, 25, 27]
```

II .COURSE OUTCOME 2(CO2)

PROGRAM NO: 1

DATE:1/12/2021

AIM: Program to find the factorial of a number

PROGRAM

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

OUTPUT

```
Enter the number:5
Factorial of 6 is: 120
```

PROGRAM NO: 2

DATE:1/12/2021

AIM: Generate Fibonacci series of N terms

PROGRAM

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

OUTPUT

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

PROGRAM NO: 3

DATE:1/12/2021

AIM: Find the sum of all items in a list

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

total = sum(list1)

print(“sum of list:”,total)
```

OUTPUT

```
Sum of list : 100
```

PROGRAM NO: 4

DATE:1/12/2021

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

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

OUTPUT

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

PROGRAM NO:5

DATE:1/12/2021

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

```
rows = int(input("Enter the number of rows: "))

fori in range(1, rows+1):

for j in range(1,i+1):

print(i * j, end=' ')

print()
```

OUTPUT

```
Enter the number of rows:
1

2 4

3 6 9

4 8 12 16
```

PROGRAM NO:6

DATE:1/12/2021

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

PROGRAM

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

OUTPUT

```
Enter the string:malayalam

{'m': 2, 'a': 4, 'l': 2, 'y': 1}
```

PROGRAM NO:7

DATE:8/12/2021

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

PROGRAM

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

OUTPUT

```
Enter the string:play
Inputted string: play
Formatted string: playing

Enter the string:coming
Inputted string: coming
Formatted string comingly
```

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:2
Enter element 1 python
Enter element 2 programming
Longest Word: programming
Length of longest word : 11
```

PROGRAM NO:9

DATE:8/12/2021

AIM: Construct following pattern using nested loop

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

PROGRAM

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

OUTPUT

```
Enter the limit:4

*

* *

* * *

* *

*
```

PROGRAM NO:10

DATE:8/12/2021

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

PROGRAM

```
 def fact(n):

 print("Factors of",n,":")

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

  if n%i==0:

   print(i)

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

fact(n)
```

OUTPUT

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

PROGRAM NO:11

DATE:8/12/2021

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

PROGRAM

```
a_sq=lambda a:a*a

a_rec=lambda l,b:l*b

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


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

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

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

OUTPUT

```
Area of square= 4
Area of rectangle= 4
Area of triangle= 5.0
```

III .COURSE OUTCOME 3(CO3)

PROGRAM NO: 1

DATE:15/12/2021

AIM: Work with built-in packages

<u>Time Module</u>

PROGRAM

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

OUTPUT

```
Current time in sec: 1639915265.630671
Current time: Mon Dec 19 17:31:05 2021
Time After 30 sec: Mon Dec 19 17:31:35 2021

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)
current year: 2021
current month 12
current day 19
current hour 17
current minute 31
current sec 5
current week day 6
current year day 353
```

Math module

PROGRAM

```
import math
print(math.pi)

import math as m
print(m.pi)

from math import pi,sqrt
print(math.pi)
print(math.sqrt(4))
print(math.cos(90))
print(math.sin(1/2))
print(math.tan(0))
```

OUTPUT

```
 3.141592653589793
 3.141592653589793
 3.141592653589793
 2.0
-0.4480736161291701
 0.479425538604203
 0.0
```

Calendar module

PROGRAM

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

OUTPUT

```
Enter month: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

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(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

```
 22:56:44
Hour 22
Minutes 56
Seconds 44
Microsecond 0
2021-12-20
Year: 2021
Month: 12
Day: 20
2021-12-20
2 days, 0:00:00
2021-12-22
2021-12-20 22:56:44
```

Statistics module

PROGRAM

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

OUTPUT

```
5.461538461538462
6
7
2.569545505058064
6.602564102564102
```

Random module

PROGRAM

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

OUTPUT

```
12
0.23604808973743452
0.1031660342307158
3
```

PROGRAM NO: 2

DATE:15/12/2021

AIM:

Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

- **graphics package**

circle module

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

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

rectangle module

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

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

- **3dgraphics package**

sphere module

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

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

cuboid module

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

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

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

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

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

print("\nCuboid:")

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

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

OUTPUT

```
Rectangle:
Enter the length:2
Enter the breadth:2
Area= 4
Perimeter= 8

Circle:
Enter the radius:2
Area= 12.56
Perimeter= 12.56

Cuboid:
Enter the length:2
Enter the width:2
Enter the height:1
Enter the breadth:2
Area= 16
perimeter= 20
Sphere:
Enter the radius:2
Area= 50.24
perimeter= 12.56
```

IV .COURSE OUTCOME 4(CO4)

PROGRAM NO: 1

DATE:9/1/2022

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

PROGRAM

```
class Rectangle:
    def __init__(self,l,b):
    self.l=l
    self.b=b
    self.area=self.l*self.b
    self.p=2*(self.l+self.b)
    def display(self):
     print("Area:",self.area)
     print("Perimeter:",self.p)
p1=Rectangle(3,2)
p2=Rectangle(2,4)
print("R1")
p1.display()
print("R2")
p2.display()

if p1.area>p2.area:
    print(" Rectangle with area ",p1.area,"is larger")
else:
    print(" Rectangle with area ",p2.area,"is larger")
```

OUTPUT

```
R1
Area:6
Perimeter:10
R2
Area:8
Perimeter= 12

Rectangle with area 8 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.

PROGRAM

```
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:12345678
Enter the name:Devu
Enter the account type-(savings/current):savings
Enter the initial balance:2000

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:2
Enter the amount to deposit:2000

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

Account Info:
```

```
Account Number: 1235678
Account Name: Devu
Account Type: savings
Account Balance: 4000

1.Account Info
2.Deposit
3.Withdraw
4.Exit
Select your option:3
Enter the amount to withdraw:1000
RS- 1000 Withdrawn successfully

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

Account Info:
Account Number: 12345678
Account Name: Devu
Account Type: savings
Account Balance: 3000

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

PROGRAM NO: 3

DATE:9/1/2022

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

PROGRAM

```
 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:6
Enter the breadth:7
Area= 42

Second Rectangle:
Enter the length:4
Enter the breadth:9
Area= 36

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

PROGRAM

```
class Time:
    def __init__(self,hour,minute,second):
        self.__hour=hour
        self.__minute=minute
        self.__second=second
    def __add__(self,a2):
        second=self.__second+a2.__second
        minute=self.__minute+a2.__minute
        hour=self.__hour+a2.__hour
        if(second>60):
           second=second-60
           minute=minute+1
       if(minute>60):
           minute=minute-60
           hour=hour+1
       return hour,minute,second
print("Enter time1:")
h1=int(input("hour:"))
m1=int(input("minute:"))
s1=int(input("second"))

t1=Time(h1,m1,s1)

print("Enter time2:")
h2=int(input("hour:"))
m2=int(input("minute:"))
s2=int(input("second"))

t2=Time(h2,m2,s2)

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

OUTPUT

```
Enter time1:
 hour:12
 minute:27
 second38
 Enter time2:
 hour:11
 minute:45
 second23
 24:13:1
```

PROGRAM NO:5

DATE:9/1/2022

AIM:

Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

PROGRAM

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


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

OUTPUT

```
Title : Python Programming
Author : John Kennady
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.

PROGRAM

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

OUTPUT

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

PROGRAM NO: 2

DATE:30/1/2022

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

PROGRAM

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

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

OUTPUT

```
This is my first file in python.

Want to work with files.

This is my third

Want to work with files.
```

PROGRAM NO: 3

DATE:30/1/2022

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

PROGRAM

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

username.csv

```
Username;Identifier;Firstname:Lastname

Booker12;9012;Rachel;Booker

Grey07;2070;Laura;Grey

Johnson81;4081;Craig;Johnson

Jenkins46;9346;Mary;Jenkins

Smith79;5079;Jamie;Smith
```

OUTPUT

```
[['Username;Identifier;Firstname;Lastname'],

['booker12;9012;Rachel;Booker'],['grey07;2070;Laura;Grey'],

['johnson81;4081;Craig;Johnson'],['jenkins46;9346;Mary;Jenkins'],

['smith79;5079;Jamie;Smith']]
```

PROGRAM NO: 4

DATE:30/1/2022

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

PROGRAM

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

emp.txt

```
name,      department,    birthday month
John Smith,  Accounting,   November
Erica Meyers, IT,          March
```

OUTPUT

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

PROGRAM NO: 5

DATE:30/1/2022

AIM: Write a Python program to write a Python dictionary to 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)
filename = "names1.csv"
cf=open("names1.csv", 'r')
rows=[]
csvreader = csv.reader(cf)

for r in csvreader:
  rows.append(r)
for r in rows:
      print(*r)
```

cars.csv

```
No,Company,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,Model

 1 Ferrari,488, GTB

 2 Porsche,918, Spyder

 3 Bugatti,La, Voiture, Noire

4 Rolls Royce Phantom

5 BMW BMW X7
```