20MCA131 PROGRAMMING LAB

Lab Report Submitted By:

JUSTIN.V.KALAPPURA
Reg. No.: AJC21MCA-2068

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS (2 Year)
(MCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovapally, Kanjirappally, Kottayam, Kerala – 686518]

2021-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

CERTIFICATE

This is to certify that the lab report, "**20MCA131 PROGRAMMING LAB"** is the bonafide work of **JUSTIN.V.KALAPPURA (AJC21MCA-2068)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Sruthimol kurian
Staff In-Charge

CONTENT

17	Merge two dictionaries.		23
18	Find GCD of 2 numbers.		24
19	From a list of integers, create a list removing even numbers.		25
20	Program to find the factorial of a number.		26
21	Generate Fibonacci series of N terms.		27
22	Find the sum of all items in a list.		28
23	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.		29
24	Display the given pyramid with step number accepted from user.		30
25	Count the number of characters (character frequency) in a string.		31
26	Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.		32
27	Accept a list of words and return length of longest word.		33
28	Construct pattern using nested loop.		34
29	Generate all factors of a number.		35
30	Write lambda functions to find area of square, rectangle and triangle.		36
31	Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere.		37-38
32	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.		39
33	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.		40-41
34	Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.		42
35	Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.		43

36	Create a class for Book Publisher and performance inheritance.		44
37	Write a Python program to read a file line by line and store it into a list.		45
38	Program to copy odd lines of one file to other.		46
39	Write a Python program to read each row from a given csv file and print a list of strings.		47
40	Write a Python program to read specific columns of a given CSV file and print the content of the columns.		48
41	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.		49

Program no:1

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

Source Code:

```
start_yr=int(input("Enter the starting year\n"))
limit_yr=int(input("Enter the limit year\n"))

lst=[]

for i in range(start_yr,limit_yr):
   if((i%4==0 and i%100!=0) or i%400==0):
      lst.append(i)

print("leap year",lst)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
r.py"
Enter the starting year
2000
Enter the limit year
2020
leap year [2000, 2004, 2008, 2012, 2016]
```

Program no:2

Aim: List comprehensions:

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

b. Square of N numbers

c. Form a list of vowels selected from a given word

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

Source Code:

```
lst = [10,-79,34,12,-90,56]
lstnew = [n for n in lst if n >= 0]
print("list:",lst)
print("New List Is:", lstnew)

num=int(input("enter the limit\t"))
sqr=[i*i for i in range(num+1)]
print("square of numbers is\t",sqr)

word=input("enter the word\t")
count = {}.fromkeys(['a','e','i','o','u'], 0)
for c in word.casefold():
      if c in count:
         count[c] += 1
print(count)

myinput=input("enter the message\t")
mylist =list(myinput)
mylist = [ord(character) + 1 for character in mylist]
print(mylist)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
list: [10, -79, 34, 12, -90, 56]
New List Is: [10, 34, 12, 56]
```

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the limit 5
square of numbers is     [0, 1, 4, 9, 16, 25]
```

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the word  python world
{'a': 0, 'e': 0, 'i': 0, 'o': 2, 'u': 0}
```

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the message       python
[113, 122, 117, 105, 112, 111]
```

Program no:3

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

Source Code:

```
words=input("enter the sentance\t")
count={}.fromkeys(words.casefold().split(' '),0)
print(count)
for i in words.casefold().split(' '):
   count[i]+=1
print(count)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> python -u "d:\clg lab
rance.py"
enter the sentance      hlo python welcom to the world of python
{'hlo': 0, 'python': 0, 'welcom': 0, 'to': 0, 'the': 0, 'world': 0, 'of': 0}
{'hlo': 1, 'python': 2, 'welcom': 1, 'to': 1, 'the': 1, 'world': 1, 'of': 1}
```

Program no: 4

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

Source Code:

```
lst=[]
l=int(input("enter the limit\t"))
for i in range(l):
    a=int(input("enter the value\t"))
    lst.append('over' if a>100 else a)
print(lst)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the limit 5
enter the value 34
enter the value 120
enter the value 67
enter the value 560
enter the value 33
[34, 'over', 67, 'over', 33]
```

Program no: 5

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

Source Code:

```
l=int(input("enter the num of names\t"))
counts=0
for i in range(l):
    a=input("enter the name\t").split(" ")[0]
    counts+=a.count('a')
print(counts)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the num of names  3
enter the name  python
enter the name  java
enter the name  django
3
```

Program no: 6

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

Source Code:

```
string1=input("enter the string\t")
b=0
ch=string1[0]
for c in string1[1:-1]:
   if c==ch:
      b=string1.replace(c,'$')
print(ch+b[1:])

import math
rad=int(input("enter the radius\t"))
print("radius is:",math.pi*(rad*rad))

lst=[235,'hello',890,'new','90',123,'mrng']
print("list is:\t",lst)
for i in range(len(lst),0,-1):
   print(lst[i-1])
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyt
[2, 5, 7, 8, 9, 1, 2, 3] [5, 8, 9, 1, 2, 3, 4, 7, 8, 0, 2]
8 11
length is not same

PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyt
[2, 5, 7, 8, 9, 1, 2, 3] [5, 8, 9, 1, 2, 3, 4, 7, 8, 0, 2]
sum is not same
sum is list1 37 and sum of list 2 is49

PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> python -u "d:\c
RunnerFile.py"
[2, 5, 7, 8, 9, 1, 2, 60, 70, 20] [5, 8, 9, 1, 2, 3, 674, 7, 8, 0, 200]
value occurance in both list is:         [2, 5, 7, 8, 9, 1, 2]
```

Program no: 7

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

Source Code:

```
str1=input('enter the string 1\t')
b=0
ch=str1[0]
for c in str1[1:-1]
   if c==ch;
    b=str1.replace(c,'$')
print(ch+b[1:])
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the string        onion
oni$n
```

Program no: 8

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

Source Code:

```
word='python'
print("Before Swapping\t",word)
print("After Swapping\t",word[-1]+word[1:-1]+word[0])
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
ng.py"
Before Swapping  python
After Swapping   nythop
```

Program no: 9

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

Source Code:

```
import math
rad=int(input("enter the radius\t"))
print("radius is:",math.pi*(rad*rad))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the radius        5
radius is: 78.53981633974483
```

Program no: 10

Aim: Find biggest of 3 numbers entered.

Source Code:

```
a=int(input("enter the 1st number\n"))
b=int(input("enter the 2nd number\n"))
c=int(input("enter the 3rd number\n"))

if(a>b and a>c):
   print(f"{a} is greater\n")
elif(b>c):
   print(f"{b} is greater\n")
else:
   print(f"{c} is greater")
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> 
ofthreenum.py"
enter the 1st number
45
enter the 2nd number
21
enter the 3rd number
22
45 is greater
```

Program no: 11

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

Source Code:

```
filename=input("enter the filename\t")
ext=filename.split(".")[1]
print(ext)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the filename      demo.java
java
```

Program no: 12

Aim: Create a list of colors from comma-separated color names entered by user. Display first and last colors.

Source Code:

```
n=int(input("enter the number\t"))
lst=[]
for i in range(n):
   lst.append(input("enter the color\t"))

print('first color is {} last color is {}'.format(lst[0],lst[-1]))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the number        5
enter the color red
enter the color blue
enter the color green
enter the color yellow
enter the color black
first color is red last color is black
```

Program no: 13

Aim: Accept an integer n and compute n+nn+nnn.

Source Code:

```
n=int(input("enter the number"))
temp=str(n)
t1=temp+temp
t2=temp+temp+temp
cal=n+int(t1)+int(t2)
print(cal))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
y"
enter the number3
369
```

Program no: 14

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

Source Code:

```
color1 = ["White", "Black", "Red"]
color2 = ["Red", "green"]
print("Colour list is")
print(color1)
print(color2)
print("colours not in 2nd list is:\t")
for z in color1:
   if z not in color2:
      print(z)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
RunnerFile.py"
Colour list is
['White', 'Black', 'Red']
['Red', 'green']
colours not in 2nd list is:
White
Black
```

Program no: 15

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

Source Code:

```
str1=input('enter the string 1\t')

str2=input('enter the 2nd string\t')

print(str2[0]+str1[1:]+' '+str1[0]+str2[1:])
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the string 1      python
entetr the 2nd string   java
jython pava
```

Program no:16

Aim: Sort dictionary in ascending and descending order.

```
print("Dictionary Sorting")
dict1={'Milan':43,'Raju':12,'Renju':90,'Biju':16,'Bibin':210,'Arjun':190}
print("Sorting of dictionary")
print(dict1)
print(sorted(dict1.items()))
print("Ascending order",sorted(dict1.items(),key=lambda x:x[1]))
print("Descending order",sorted(dict1.items(),key=lambda x:x[1],reverse=True))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> python -u "d:\clg labs\python prgrm\programming Lab\la
RunnerFile.py"
Dictionary Sorting
Sorting of dictionary
{'Milan': 43, 'Raju': 12, 'Renju': 90, 'Biju': 16, 'Bibin': 210, 'Arjun': 190}
[('Arjun', 190), ('Bibin', 210), ('Biju', 16), ('Milan', 43), ('Raju', 12), ('Renju', 90)]
Ascending order [('Raju', 12), ('Biju', 16), ('Milan', 43), ('Renju', 90), ('Arjun', 190), ('Bibin', 210)]
Descending order [('Bibin', 210), ('Arjun', 190), ('Renju', 90), ('Milan', 43), ('Biju', 16), ('Raju', 12)]
```

Program no: 17

Aim: Merge two dictionaries.

Source Code:

```
dict1 = {'a': 10, 'b': 8}
dict2 = {'d': 6, 'c': 4}

dict2.update(dict1)

print(dict2)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
rge.py"
{'d': 6, 'c': 4, 'a': 10, 'b': 8}
```

Program no: 18

Aim: Find gcd of 2 numbers.

Source Code:

```
num1=int(input("enter the 1st number\n"))
num2=int(input("enter the 2nd number\n"))
key=num1 if num1<num2 else num2
hcf=1
for i in range(key,0,-1):
    if num1%i==0 and num2%i==0:
        hcf=i
        break
print(f" The hcf of the numbers {num1} and {num2} is {hcf}")
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>

enter the 1st number
20
enter the 2nd number
10
 The hcf of the numbers 20 and 10 is 10
```

Program no: 19

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

Source Code:

```
num = [1,4,6,9,2,34,29,47,12,43,44,90]
print("number is:",num)
print("non-even number is:")

num = [a for a in num if a%2!=0]
print(num)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
.py"
number is: [1, 4, 6, 9, 2, 34, 29, 47, 12, 43, 44, 90]
even number is:
[1, 9, 29, 47, 43]
```

Program no: 20

Aim: Program to find the factorial of a number

Source Code:

```
def fact1(num):
    fact=1
    if num==0:
        return 0
    elif num==1:
        return 1
    else:
        for i in range(1,num+1):
            fact=fact*i
        return fact
num=int(input("enter the number\t"))
print("factorial is {0}".format(fact1(num)))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the number        5
factorial is 120
```

Program no: 21

Aim: Generate Fibonacci series of N terms

Source Code:

```
def fib(num):
  a=0
  b=1
  sum=0
  count=1
  lst=[]
  while(count<=num):
    lst.append(sum)
    count+=1
    a=b
    b=sum
    sum=a+b
  return lst

num = int(input("Enter the value of limit \t"))
print("Fibonacci Series: {0}".format(fib(num)))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
Enter the value of limit        5
Fibonacci Series: [0, 1, 1, 2, 3]
```

Program no: 22

Aim: Find the sum of all items in a list.

Source Code:

```
lst=[3,4,5,6,1,2,8,9,34,56,12]
print(lst)
sum=0
for i in lst:
   sum+=i
print(sum)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
[3, 4, 5, 6, 1, 2, 8, 9, 34, 56, 12]
140
```

Program no: 23

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

Source Code:

```
l=int(input("enter the lower limit\t"))
u=int(input("enter the higher limit\t"))
lst=[x for x in range(l,u+1) if x**.5== int(x**.5)]
lst=[x for x in lst if x%2==0 ]
print(lst)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the lower limit   1000
enter the higher limit  1200
[1024, 1156]
```

Program no: 24

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

Eg: N=4

1

2 4

3 6 9

4 8 12 16

Source Code:

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

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the number        5
1

2 4

3 6 9

4 8 12 16

5 10 15 20 25
```

Program no: 25

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

Source Code:

```
lst="python java"
lstnew=[x for x in lst.casefold()]
dict={}.fromkeys(lstnew,0)
for x in lst.casefold():
    if x in dict:
        dict[x]+=1
print(lst)
print(dict)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> python -u "d:\clg labs\py
python java
{'p': 1, 'y': 1, 't': 1, 'h': 1, 'o': 1, 'n': 1, ' ': 1, 'j': 1, 'a': 2, 'v': 1}
```

Program no: 26

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

Source Code:

```
a=input("enter the word\t")

if a[-3:]=='ing':
   print(a+'ly')
elif a[-3:]!='ing':
   print(a+'ing')
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the word  sing
singly
```

Program no: 27

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

Source Code:

```
lst=['hello','world','python','java','newpython','jupyter']
a=0
print(lst)
for x in lst:
   a= len(x) if len(x)>a else a
print(a)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyt
['hello', 'world', 'python', 'java', 'newpython', 'jupyter']
9
```

Program no: 28

Aim: Construct following pattern using nested loop

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

Source Code:

```
a=int(input("enter the limit\t"))
for i in range(1,a+1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")

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

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the limit 5
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

Program no: 29

Aim: Generate all factors of a number.

Source Code:

```
num=int(input("enter the number\t"))
lst=[i for i in range(1,num+1) if num%i==0]
print(lst)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the number        10
[1, 2, 5, 10]
```

Program no: 30

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

Source Code:

```
import math
x=int(input("enter the length of the square of 1 side\t"))
area=lambda x:math.pi*x
print("area of square is {}".format(area(x)))

l=int(input("enter the length of the rectangle\t"))
b=int(input("enter the breadth of the rectangle\t"))
area=lambda l,b:l*b
print("area of rectangle is {}".format(area(l,b)))

b=int(input("enter the breath of the triangle\t"))
h=int(input("enter the height of the triangle\t"))
area=lambda b,h:0.5*b*h
print("area of the triangle is {}".format(area(b,h)))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the length of the square of 1 side        5
area of square is 15.707963267948966
enter the length of the rectangle       5
enter the breadth of the rectangle      10
area of rectangle is 50
enter the breath of the triangle        10
enter the height of the triangle        12
area of the triangle is 60.0
```

Program no: 31

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).

Source Code:

module.py

```
import math

def rectangle(length, width):
   area=length*width
   perimeter=2*(length+width)

   return "Area is {} and Perimeter is {}".format(area,perimeter)

def circle(r):
   area=math.pi*(r**2)
   perimeter=2*math.pi*r

   return "Area is {} and Perimeter is {}".format(area,perimeter)

def cuboid(length, breadth,height):
   area=2*((length*breadth)+(breadth+height)+(height*length))
   perimeter=4*(length+breadth+height)

   return "Area is {} and Perimeter is {}".format(area,perimeter)

def sphere(r):
   area=2*r
   perimeter=4*math.pi*(r**2)

   return "Area is {} and Perimeter is {}".format(area,perimeter)
```

main.py

```
from module import *

# Rectangle
length=float(input("enter the length\t"))
breadth=float(input("enter the width\t"))
print(rectangle(length,breadth))
```

```
# circle
radius=float(input("enter the radius\t"))
print(circle(radius))


# cuboid
length=float(input("enter the length\t"))
breadth=float(input("enter the breadth\t"))
height=float(input("enter the height\t"))
print(cuboid(length, breadth,height))

# sphere
radius=float(input("enter the radius\t"))
print(sphere(radius))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyt
enter the length        10
enter the width 20
Area is 200.0 and Perimeter is 60.0
enter the radius        5
Area is 78.53981633974483 and Perimeter is 31.41592653589793
enter the length        10
enter the breadth       5
enter the height        9
Area is 308.0 and Perimeter is 96.0
enter the radius        10
Area is 20.0 and Perimeter is 1256.6370614359173
```

Program no: 32

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

Source Code:

```
class Rectangle:
    def __init__(self,length, breadth):
        self.length = length
        self.breadth = breadth

    def area(self):
        return self.length*self.breadth

    def perimeter(self):
        return 2*(self.length + self.breadth)


l1=float(input("Enter length of rectangle: "))
b1=float(input("Enter breadth of rectangle: "))
l2=float(input("Enter length of rectangle: "))
b2=float(input("Enter breadth of rectangle: "))
rect1=Rectangle(l1,b1)
rect2=Rectangle(l2,b2)
print("Area of rectangle 1 is {} and perimeter is {}: ".format(rect1.area(),rect1.perimeter()))
print("Area of rectangle 2 is {} and perimeter is {}: ".format(rect2.area(),rect2.perimeter()))
print(rect1.area()>rect2.area())
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
Enter length of rectangle: 10
Enter breadth of rectangle: 20
Enter length of rectangle: 5
Enter breadth of rectangle: 30
Area of rectangle 1 is 200.0 and perimeter is 60.0:
Area of rectangle 2 is 150.0 and perimeter is 70.0:
True
```

Program no: 33

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.

Source Code:

```
class Bank:
    def __init__(self, account_number, name, account_type, balance):
        self.account_number = account_number
        self.name = name
        self.account_type = account_type
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        print("Deposit of {} successful".format(amount))
        print("Current balance is {}".format(self.balance))

    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance")
        else:
            self.balance -= amount
            print("Withdrawal of {} successful".format(amount))
            print("Current balance is {}".format(self.balance))

num=int(input("Enter account number: "))
name=input("Enter name: ")
acctype=input("Enter account type: ")
bal=int(input("Enter balance: "))
bnk=Bank(num,name,acctype,bal)
print("Account number: ",bnk.account_number)
print("Name: ",bnk.name)
print("Account type: ",bnk.account_type)
print("Balance: ",bnk.balance)
bnk.withdraw(int(input("Enter amount to withdraw: ")))
bnk.deposit(int(input("Enter amount to deposit: ")))
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
Enter account number: 1023456
Enter name: demo
Enter account type: savings
Enter balance: 50000
Account number:  1023456
Name:  demo
Account type:  savings
Balance:  50000
Enter amount to withdraw: 2000
Withdrawal of 2000 successful
Current balance is 48000
Enter amount to deposit: 5000
Deposit of 5000 successful
Current balance is 53000
```

Program no: 34

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

Source Code:

```
class Rectangle:
    def __init__(self, length, width):
        self.__length = length
        self.__width = width
        self.area=length*width

    def __lt__(self, other):
        if self.area<other.area:
            return "Reactangle 1 is smaller in Area"
        else:
            return "Reactangle 2 is smaller in Area"
r1=Rectangle(50,20)
r2=Rectangle(30,10)
print(r1<r2)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
Reactangle 2 is smaller in Area
```

Program no: 35

Aim: Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

Source Code:

```
class Time:
    def __init__(self, hour, minute, second):
        self.__hour = hour
        self.__minute = minute
        self.__second = second

    def __add__(self, other):
        return 'time is: ' + str(self.__hour + other.__hour) + ':' + str(self.__minute +
other.__minute) + ':' + str(self.__second + other.__second)


h=int(input("enter the hour: "))
m=int(input("enter the minute: "))
s=int(input("enter the second: "))
h1=int(input("enter the hour: "))
m1=int(input("enter the minute: "))
s1=int(input("enter the second: "))
t1=Time(h,m,s)
t2=Time(h1,m1,s1)
print(t1+t2)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
enter the hour: 10
enter the minute: 25
enter the second: 20
enter the hour: 5
enter the minute: 30
enter the second: 10
time is: 15:55:30
```

Program no: 36

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.

Source Code:

```
class Publisher:
    def __init__(self, Pubname):
        self.Pubname = Pubname
    def display(self):
        print("Publisher name is:", self.Pubname)
class Book(Publisher):
    def __init__(self, Pubname, title, author):
        Publisher.__init__(self, Pubname)
        self.title = title
        self.author = author
    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
class Python(Book):
    def __init__(self, Pubname, title, author, price, no_of_pages):
        Book.__init__(self, Pubname, title, author)
        self.price = price
        self.no_of_pages = no_of_pages
    def display(self):
        print("Title:", self.title)
        print("Author:", self.author)
        print("Publisher:",self.Pubname)
        print("Price:", self.price)
        print("No of pages:", self.no_of_pages)
b1 = Python("New India", "Python For Babies", "Mark Liyo", 600, 900)
b1.display()
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs>
Title: Python For Babies
Author: Mark Liyo
Publisher: New India
Price: 600
No of pages: 900
```

Program no: 37

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

Source Code:

```
file=open('demo.txt')
lst=[]
for line in file:
   lst.append(line)
print(lst)
file.close()
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> python
['python \n', 'Django \n', 'flask\n', 'Java \n', 'Javascript']
```

Program no: 38

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

Source Code:

```
newFile = open("States.txt","w")
newFile.write("Goa \nKerala \nTamilnadu \nBangal\nPunjab")
newFile.close()

readFile = open("States.txt","r")
lines = readFile.readlines()
print(lines)
readFile.close()

oddFile = open("oddcontent.txt","w")
for i in range(0,len(lines),2):
    oddFile.write(lines[i])
oddFile.close()

readFile = open("oddcontent.txt","r")
lines = readFile.readlines()
print(lines)
readFile.close()
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs> pyth
['Goa \n', 'Kerala \n', 'Tamilnadu \n', 'Bangal\n', 'Punjab']
['Goa \n', 'Tamilnadu \n', 'Punjab']
```

Program no: 39

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

Source Code:

```
import csv
with open('emp.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["ID", "Name", "Age","Salary"])
    writer.writerow([121, "Revathy S", "26","30000"])
    writer.writerow([122, "Arun Kumar", "24","30000"])
with open('emp.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs\extra>
['ID', 'Name', 'Age', 'Salary']
['121', 'Revathy S', '26', '30000']
['122', 'Arun Kumar', '24', '30000']
```

Program no: 40

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

Source code:

```
import csv
with open('employee.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["ID", "Name", "Age", "Salary"])
    writer.writerow([121, "Revathy S", "26", "30000"])
    writer.writerow([122, "Arun Kumar", "24", "30000"])
with open('employee.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row[1] + " " + row[2])
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs\extra>
Name Age
Revathy S 26
Arun Kumar 24
```

Program no: 41

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.

Source code:

```
import csv
f = open("planets.csv","w",newline='')
writer = csv.DictWriter(f, fieldnames = ["Planet","count"])
writer.writeheader()
writer.writerow({"Planet":"Mercury", "count": "1"})
writer.writerow({"Planet": "Venus", "count": "2"})
writer.writerow({"Planet": "Earth", "count": "3"})
f.close()
c = 0
f = open("planets.csv")
reader = csv.DictReader(f)
for row in reader:
    if c==0:
        print(f'{"".join(row)}')
        c+=1
    print(f'{row["Planet"]},{row["count"]}')
    c+=1
print(c-1)
f.close()
```

Output:

```
PS D:\clg labs\python prgrm\programming Lab\lab-git\labs\extra>
Planetcount
Mercury,1
Venus,2
Earth,3
3
```