

PRACTICAL NO :1

A.Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

CODE:

```
from datetime import date
Name=str(input("Enter your Name"))
age=int(input("Enter your age"))
current=date.today()
year_after=current.year+(100-age)
print("hii",Name,"your cureent age is ",age," and you will complete 100 years
in",year_after)
```

OUTPUT:

```
Enter your NameSejal
Enter your age20
hii sejal your cureent age is 20 and you will complete 100 years in 2104
```

B.Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.

CODE:

```
number=int(input("Enter your number"))
if number%2==0:
    print("Your number is EVEN")
else:
    print("Your number is ODD")
```

OUTPUT:

```
Enter your number12
Your number is EVEN
```

```
Enter your number45
Your number is ODD
```

c. Write a program to generate the Fibonacci series.

CODE

```
num = 10
n1=0
n2 = 1
print("Fibonacci Series:", n1, n2, end=" ")
for i in range(2, num):
    n3 = n1 + n2
    n1 = n2
    n2 = n3
    print(n3, end=" ")
```

OUTPUT:

Fibonacci Series: 0 1 1 2 3 5 8 13 21 34

d. Write a function that reverses the user defined value.

CODE:

```
def reverr(num):
    temp=num
    rev=0
    while temp>0:
        digit=temp%10
        rev=rev*10+digit
        temp=temp//10
    print(rev)
    num=int(input("ENTER A NUMBER : "))
    reverr(num)
```

CODE: ENTER A NUMBER : 123
321

e. Write a function to check the input value is Armstrong and also write the function for Palindrome.

CODE:

```
def palindrm(num):
    temp=num
    pal=0
    while temp>0:
        digit=temp%10
        pal=pal*10+digit
        temp=temp//10
    if pal==num:
        print("number is palindrome")
    else:
        print("number is not palindrome")
num=int(input("ENTER A NUMBER : "))
palindrm(num)
def armstr(num):
    length=len(str(num))
    temp=num
    arm=0
    while temp>0:
        digit=temp%10
        arm=arm+digit**length
        temp=temp//10
    if arm==num:
        print("number is Armstrong")
    else:
        print("number is not Armstrong")
num=int(input("ENTER A NUMBER : "))
armstr(num)
```

OUTPUT: ENTER A NUMBER : 121

number is palindrome
 ENTER A NUMBER : 153
 number is Armstrong

ENTER A NUMBER : 456
 number is not palindrome
 ENTER A NUMBER : 354
 number is not Armstrong

f. Write a recursive function to print the factorial for a given number.

CODE:

```
def factorial(n):
    if (n==1 or n==0):
        return 1
    else :
        return n * factorial(n - 1)
    return factorial(num)
num =int(input("enter number"))
print("Factorial of",num,"is",factorial(num))
```

OUTPUT: enter number5
 Factorial of 5 is 12

Practical No: 02**Write the program for the following:**

- a. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.

Code:

```
def vowels(character):
    vowel=["a","e","i","o","u","A","E","I","O","U"]
    if character in vowel:
        print("The character is a vowel")
    else:
        print("The character is not a vowel")
character=input("Enter a character: ")
vowels(character)
```

Output

Enter a character: O
The character is a vowel

Enter a character: R
The character is not a vowel

- b. Define a function that computes the length of a given list or string.

Code:

```
def cal_length(ch):
    count=0
    for x in ch:
        count=count+1
    return count
a=int(input("Enter \n1.String\n2.List\n"))
if(a==1):
    ch=input("Enter your string: ")
    print("Length of your string is: ",cal_length(ch))
elif(a==2):
    li=[]
    print("If you want to stop than enter 'stop'...")
    for i in range(100):
        el=input("Enter your list: ")
        if(el=='stop'):
            print("Length of your list is: ",cal_length(li))
            break;
        else:
            li.append(el)
```

Output:

Enter
1.String
2.List
1
Enter your string: data structure
Length of your string is: 14

Enter
1.String
2.List
2
If you want to stop than enter 'stop'...
Enter your list: 45
Enter your list: 63
Enter your list: 93
Enter your list: 18
Enter your list: 17
Enter your list: stop
Length of your list is: 5

- c. Define a *procedure* histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:

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

Code:

```
def histogram():
    l=[]
    a=int(input("Enter size for histogram: "))
    for x in range(a):
        l.append(int(input("Enter element for histogram: ")))
    for i in l:
        print('*'*i)
```

Output:

```
Enter size for histogram: 3
Enter element for histogram: 4
Enter element for histogram: 9
Enter element for histogram: 7
*****
***** *
***** *
```

PRACTICAL NO 3

A. A *pangram* is a sentence that contains all the letters of the English alphabet at least once, for example: *The quick brown fox jumps over the lazy dog*. Your task here is to write a function to check a sentence to see if it is a pangram or not.

CODE:

```
def pangram(str):
    alphabet = "abcdefghijklmnopqrstuvwxyz"
    for char in alphabet:
        if char not in str:
            return False
    return True
string = 'The quick brown fox jumps over the lazy dog'
if(pangram(string) == True):
    print(" Given string is Pangram")
else:
    print(" Given string is not Pangram")
```

OUTPUT:

Given string is Pangram

B. Take a list, say for example this one:

a=[1,1,2,3,5,8,13,21,34,55,89] and write a program that prints out all the elements of the list that are less than 5.

CODE:

```
a=[1,1,2,3,5,8,13,21,34,55,89]
l=[]
for num in a:
    if (num < 5):
        print(num)
    else:
        continue
```

OUTPUT:

1
1
2
3

Practical No: 04

Write the program for the following:

- a. Write a program that takes two lists and returns True if they have at least one common member.**

Code:

```
def common(list1,list2):
```

```
    result=False
```

```
    for x in list1:
```

```
        for y in list2:
```

```
            if(x==y):
```

```
                result=True
```

```
                return result
```

```
    return result
```

```
a=[1,2,3,4,5]
```

```
b=[5,6,7,8,9]
```

```
print(common(a,b))
```

Output:

True

- b. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.**

Code:

```
d1=[0,1,2,3,4,5,6,7,8,9]
```

```
print("The original list is: ",d1)
```

```
d2=d1.copy()
```

```
(d2.pop(0))
```

```
print("After deleting the 0th element",d2)
```

```
d3=d1.copy()
```

```
(d3.pop(2))
```

```
print("After deleting the 2th element",d3)
```

```
d4=d1.copy()
```

```
(d4.pop(4))
```

```
print("After deleting the 4th element",d4)
```

```
d5=d1.copy()
```

```
(d5.pop(5))
```

```
print("After deleting the 5th element",d5)
```

Output:

```
The original list is: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
After deleting the 0th element [1, 2, 3, 4, 5, 6, 7, 8, 9]
After deleting the 2th element [0, 1, 3, 4, 5, 6, 7, 8, 9]
After deleting the 4th element [0, 1, 2, 3, 5, 6, 7, 8, 9]
After deleting the 5th element [0, 1, 2, 3, 4, 6, 7, 8, 9]
```

c. Write a Python program to clone or copy a list.

Code:

```
a=[1,2,3,4,5]
```

```
b=[]
```

```
b=a
```

```
Print("Printed list is: ",b)
```

Output:

```
Printed list is: [1, 2, 3, 4, 5]
```

PRACTICAL NO 5

A. Write a Python script to sort (ascending and descending) a dictionary by value,

CODE:

```
u= {}
u={input("enter key"):input("value") for x in range(4)}
print("Dictionary after adding user input:", u)
l=sorted(u.items())
d={x:y for x,y in l}
print('ascending order',d)
l=sorted(u.items(),reverse=True)
d={x:y for x,y in l}
print("descending order",d)
```

OUTPUT:

```
enter key roll no
value 100
enter key name
value hari
enter key age
value 26
enter key sub
value python
Dictionary after adding user input: {'roll no': '100', 'name': 'hari', 'age': '26', 'sub': 'python'}
ascending order {'age': '26', 'name': 'hari', 'roll no': '100', 'sub': 'python'}
descending order {'sub': 'python', 'roll no': '100', 'name': 'hari', 'age': '26'}
```

B Write a Python script to concatenate following dictionaries to create a new one. Sample Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60} Expected Result : {1: 10, 2: 20, 3: 30, 4:40, 5: 50, 6: 60}

CODE:

```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
dic4=dic1|dic2|dic3
print(dic4)
```

OUTPUT:

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

PRACTICAL NO :6

a. Write a Python program to read an entire text file.

CODE:

```
f=open("my2.txt","w")
try:
    p=open("my2.txt","r")
    print(p.read())
except FileNotFoundError as e:
    print(f"error :{e}")
```

OUTPUT:

```
Key advantages of python programming
1. Readable and maintainable
2. Support multiple programming
3. Versatile, easy to read, learn, and write
4. User-friendly data structures
5. Object-Oriented and Procedural programming language
```

b. Write a Python program to append text to a file and display the text.

CODE:

```
c=input("enter text you want to write in file")
f=open("my1.txt","a")
f.write(c)
f.close()
try:
    p=open("my1.txt","r")
    print(p.read())
except FileNotFoundError as e:
    print(f"error :{e}")
```

OUTPUT:

enter text you want to write in file my first file of python
sejlkasalr
my anme
dgfdjhg
ajgxkjdg
my first file of python

c. Write a Python program to read last n lines of a file.

CODE:

```
c=int(input("which line you want to print"))
try:
    p=open("my2.txt","r")
    print(p.readlines()[c-1][0])
except FileNotFoundError as e:
    print(f"error :{e}")
```

OUTPUT:

which line you want to print4
this is file number one

Practical No:07
Write the program for the following:

- a. Design a class that store the information of student and display the same**
- Code:**

class student:

```
def __init__(self,Name,Age,City,Class):
    self.name=Name
    self.age=Age
    self.city=City
    self.classes=Class
def display(self):
    print("Student Name: ",self.name)
    print("Student Age: ",self.age)
    print("Student City: ",self.city)
    print("Student Class: ",self.classes)
```

```
s=student("Ghost",23,"Kudal","S.Y.I.T")
s.display()
```

Output:

Student Name: Ghost
 Student Age: 23
 Student City: Kudal
 Student Class: S.Y.I.T

- b. Implement the concept of inheritance using python**

1. Single inheritance:

Code:

```
class person:
    def __init__(self,Name,Age,City,mob):
        self.name=Name
        self.age=Age
        self.city=City
        self.mo=mob
    def display(self):
        print("Student Name: ",self.name)
        print("Student Age: ",self.age)
        print("Student City: ",self.city)
        print("Mobile no: ",self.mo)
class student(person):
    def __init__(self,Name,Age,City,mob,Classes):
        super().__init__(Name,Age,City,mob)
        self.classes=Classes
    def display_details(self):
        super().display()
        print("Student Class: ",self.classes)
s=student("Ghost",19,"Kudal","S.Y.I.T",9645879592)
s.display_details()
```

Output:

Student Name: Ghost
Student Age: 19
Student City: Kudal
Mobile no: S.Y.I.T
Student Class: 9645879592

2. Multiple inheritance:

Code:

```
class car_details:  
    def __init__(self,Com_name,Car_name,Fuel_type):  
        self.com_n=Com_name  
        self.car_n=Car_name  
        self.fname=Fuel_type  
    def display_car(self):  
        print("Company Name: ",self.com_n)  
        print("Company Name: ",self.car_n)  
        print("Fuel Type: ",self.fname)  
  
class register:  
    def __init__(self,Reg_no,City):  
        self.regno=Reg_no  
        self.city=City  
    def display_regi(self):  
        print("Register No: ",self.regno)  
        print("City: ",self.city)  
  
class car(car_details,register):  
    def __init__(self,Com_name,Car_name,Fuel_type,Reg_no,City,hy):  
        car_details.__init__(self,Com_name,Car_name,Fuel_type)  
        register.__init__(self,Reg_no,City)  
        self.h=hy  
    def car_print(self):  
        super().display_car()  
        super().display_regi()  
        print("Mileage: ",self.h)  
c=car("Maruti Suzuki","Swift","Petrol","MH07V1314","Kudal","15kmpl")  
c.car_detail()
```

Output:

```
Company Name: Maruti Suzuki  
Company Name: Swift  
Fuel Type: Petrol  
-----  
Register No: 101  
City: Kudal
```

1.Multilevel Inheritance:

Code:

```
class person:  
    def __init__(self,Name):  
        self.n=Name  
    def displayp(self):  
        print("Name: ",self.n)  
class student(person):  
    def __init__(self,Name,Age):  
        super().__init__(Name)  
        self.a=Age  
    def displays(self):  
        super().displayp()  
        print("Age: ",self.a)  
class emp(student):  
    def __init__(self,Name,Age,City):  
        super().__init__(Name,Age)  
        self.c=City  
    def displaye(self):  
        super().displays()  
        print("City: ",self.c)  
p=emp("Ghost",20,"Kudal")  
p.displaye()
```

OUTPUT:

```
Name: Ghost  
Age: 20  
City: Kudal
```

c. Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).

i. Write a method called add which returns the sum of the attributes x and y.

ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.

iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.

iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.

CODE:

```

class Numbers:
    MULTIPLIER=5
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def add(self):
        return self.x+self.y
    def multiply(self,a):
        return a*self.MULTIPLIER
    @staticmethod
    def subtract(b,c):
        return b-c
    @property
    def value(self):
        return self.x,self.y
    @value.setter
    def value_x(self,nx):
        self.x=nx
    @value.setter
    def value_y(self,ny):
        self.y=ny
    @value_x.deleter
    def delete_x(self):
        del self.x
    @value_y.deleter
    def delete_y(self):
        del self.y

```

```

n=Numbers(5,6)
print("Addition:- ",n.add())
print("Multiplication:- ",n.multiply(10))
print("Subtraction:- ",n.subtract(36,18))
print("Values of x and y:- ",n.value)

```

```
n.value_x=2  
n.value_y=3  
print("New Values of x and y:- ",n.value)  
n.delete_x=None  
n.delete_y=None  
print("After deleting the Values of x and y:- ",n.value)
```

OUTPUT:

Addition:- 11
Multiplication:- 50
Subtraction:- 18
Values of x and y:- (5, 6)
New Values of x and y:- (2, 3)
After deleting the Values of x and y:- (None, None)

Practical No : 8

a. Open a new file in IDLE ("New Window" in the "File" menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "Control Flow and Functions" exercise into this file and save it. Now open a new file and save it in the same directory. You should now be able to import your own module like this:

```
import geometry
```

Try and add print dir(geometry) to the file and run it.²⁶

Now write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas.

CODE:

GEOMETRY.Py

```
import math
def squareArea(side_len):
    return side_len**2;
def circleArea(radius):
    return math.pi*(radius**2);
def squarepyramid_volume(base,height):
    baseArea=squareArea(base)
    return baseArea*height/3;
def conevolume(radius,height):
    circleara=circleArea(radius)
    return circleara*height/3;
def pointyshape_volume(x,y,squarebase):
    if(squarebase):
        return squarepyramid_volume(x,y)
    else:
        return conevolume(x,y)
```

CODE:

Maths.py

```
import geometry
g=geometry.pointyshape_volume(40,36,True)
print("square pyramid value is",g)
h=geometry.pointyshape_volume(4,30,False)
print("right circular cone volume is",h)
```

OUTPUT:

```
square pyramid value is 19200.0
right circular cone volume is 502.65482457436684
```

b. Write a program to implement exception handling

CODE:

```
def calculator():
    s=[100,150,200,250,300,350,400,450,500]
    print("\n",s)
    num=int(input("select index number from the above list= "))
    div=int(input("Enter the divisor= "))
    try:
        res=s[num]/div;
        print("\n Division of ",s[num],"and ",div,"is : ",res)
    except IndexError:
        print("out of index !!")
    except ZeroDivisionError:
        print("can't divided by Zero !!")
    finally:
        ch=input("\nDo you want to continue? 1/0: ")
        if(ch.lower()=='1'):
            calculator()
calculator()
```

OUTPUT:

[100, 150, 200, 250, 300, 350, 400, 450, 500]

select index number from the above list= 9

Enter the divisor= 0

out of index !!

Do you want to continue? 1/0: 1

[100, 150, 200, 250, 300, 350, 400, 450, 500]

select index number from the above list= 11

Enter the divisor= 23

out of index !!

Do you want to continue? 1/0: 1

[100, 150, 200, 250, 300, 350, 400, 450, 500]

select index number from the above list= 2

Enter the divisor= 50

Division of 200 and 50 is : 4.0

Do you want to continue? 1/0:

Practical No: 09

Write the program for the following:

- a. Try to configure the widget with various options like: bg="red", family="times", size=18

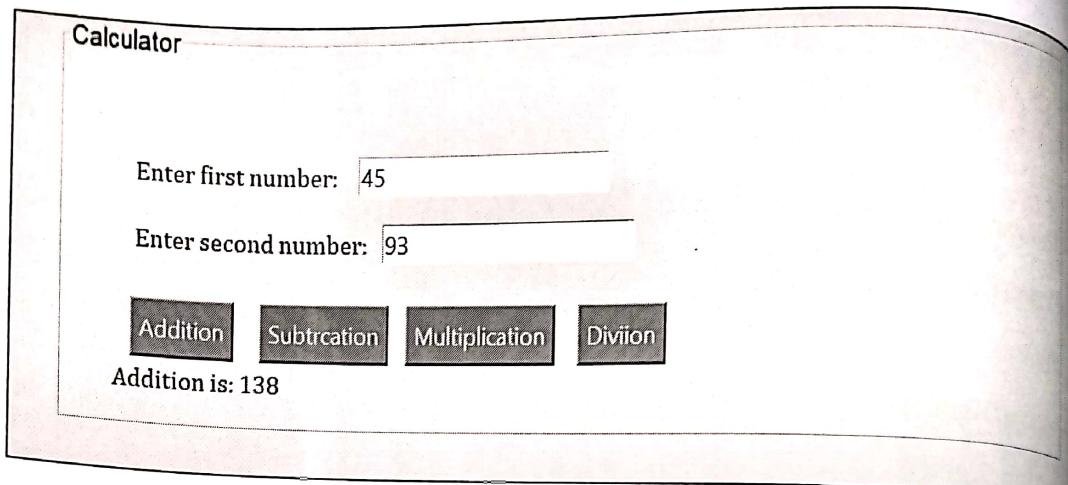
code:

```
def add():
    a=int(var1.get())
    b=int(var2.get())
    var5.set(f"Addition is: {a+b}")
def sub():
    a=int(var1.get())
    b=int(var2.get())
    var5.set(f"Subtraction is: {a-b}")
def mul():
    a=int(var1.get())
    b=int(var2.get())
    var5.set(f"Multiplication is: {a*b}")
def div():
    a=int(var1.get())
    b=int(var2.get())
    var5.set(f"Division is: {a/b}")

from tkinter import *
window=Tk()
window.geometry("800x600")
f=LabelFrame(window,text="Calculator",height=300,width=800,font="aptos")
f.pack()
var1=StringVar()
var2=StringVar()
var5=StringVar()
num1=Label(f,text="Enter first number: ",font="cambria").place(x=55,y=70)
ne1=Entry(f,textvariable=var1).place(x=240,y=70)
num2=Label(f,text="Enter second number: ",font="cambria").place(x=55,y=120)
ne2=Entry(f,textvariable=var2).place(x=260,y=120)
add=Button(f,text="Addition",command=add,bg="grey",fg="white").place(x=55,y=180)
sub=Button(f,text="Subtraction",command=sub,bg="grey",fg="white").place(x=160,y=180)
mul=Button(f,text="Multiplication",command=mul,bg="grey",fg="white").place(x=280,y=180)
div=Button(f,text="Division",command=div,bg="grey",fg="white").place(x=420,y=180)
```

```
res=Label(window,textvariable=var5,font="cambria").place(x=600,y=250)
window.mainloop()
```

output:



- b. Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale

Code:

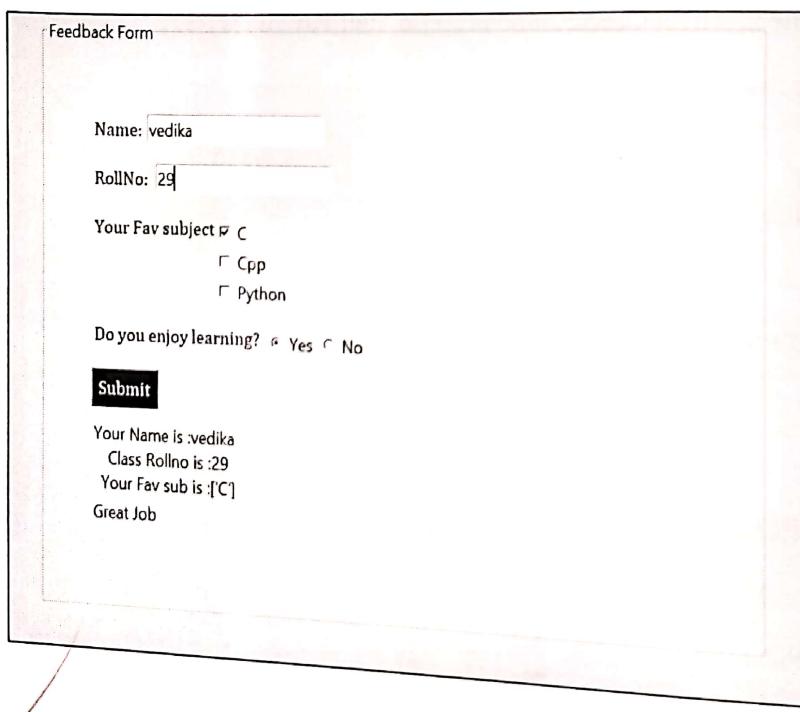
```
def show():
    n=var9.get()#for entry
    r=var10.get()#for entry
    s=[]
    if var2.get()==1: #for checkbox
        s.append("C")
    if var3.get()==1:
        s.append("Cpp")
    if var4.get()==1:
        s.append("Python")#for checkbox
    var1.set(f"Your Name is :{n}\n Class Rollno is :{r}\n Your Fav sub is :{s}\n")
    lan=var5.get() #for radio
    if lan==1:
        var6.set(f"Great Job")
    elif lan==2:
        var6.set(f"oops! It's ok ...We will try our best") #for radio
from tkinter import *
window=Tk()
window.geometry("300x300")
var1=StringVar() #for result
var2=IntVar() #for checkbox c
var3=IntVar() #for checkbox cpp
var4=IntVar() #for checkbox python
var5=IntVar() #for radio
var6=StringVar() #for result
var9=StringVar() #for result
var10=StringVar() #for result
f=LabelFrame(window,text="Feedback Form",height=600,width=800)
f.pack()
l1=Label(f,text="Name: ",font="cambria").place(x=55,y=70) #name
e1=Entry(f,textvariable=var9).place(x=120,y=70)
l2=Label(f,text="RollNo: ",font="cambria").place(x=55,y=120) #rollno
e2=Entry(f,textvariable=var10).place(x=130,y=120)
```

```

l3=Label(f,text="Your Fav subject: ",font="cambria").place(x=55,y=170) #checkbox
c=Checkbutton(f,text="C",variable=var2).place(x=200,y=170)
cpp=Checkbutton(f,text="Cpp",variable=var3).place(x=200,y=200)
python=Checkbutton(f,text="Python",variable=var4).place(x=200,y=230)
l4=Label(f,text="Do you enjoy learning?",font="cambria").place(x=55,y=280) #radiobutton
yes=Radiobutton(f,text="Yes",value=1,variable=var5).place(x=260,y=280)
no=Radiobutton(f,text="No",value=2,variable=var5).place(x=320,y=280)
btn=Button(f,text="Submit",bg="blue",fg="white",font="cambria",command=show).place(x=55,y=330) #result
result=Label(f,textvariable=var1).place(x=55,y=380)
resul=Label(f,textvariable=var6).place(x=55,y=460)
window.mainloop()

```

Output:



Practical No: 10
Design the database application for the following:

- a. Designer simple database application that stores the record and retrieve the same.
Code:

```
from customtkinter import *
import mysql.connector
con=mysql.connector.connect(host="localhost",user="root",password="123456",database="com
panies")
def ins():
    e_id=v1.get()
    e_name=v4.get()
    e_location=v2.get()
    e_sal=v3.get()
    cur=con.cursor()
    cur.execute("insert into
emp(e_name,e_location,e_sal)values(%s,%s,%s,%s)",(e_id,e_name,e_location,e_sal))
    con.commit()

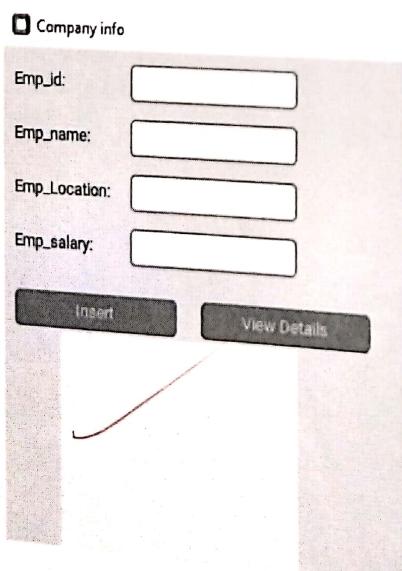
def dele():
    cur3=con.cursor()
    cur3.execute("select *from emp")
    details=cur3.fetchall()
    for i in details:
        res.insert(END,f"ID :{i[0]}\n")
        res.insert(END,f"NAME:{i[1]}\n")
        res.insert(END,f"LOCATION :{i[2]}\n")
        res.insert(END,f"SALARY:{i[3]}\n\n")
    win=CTk()
    win.title("Company info")
    win.geometry("500x500")
    win.resizable(False,False)
    set_appearance_mode("light")
    cur=con.cursor()
    v1=StringVar()
    v2=StringVar()
    v3=StringVar()
    v4=StringVar()
```

S.Y.I.T

```
id_lbl=CTkLabel(win,text="Emp_id:").place(x=10,y=10)
id_entry=CTkEntry(win,textvariable=v1,placeholder_text="id:").place(x=110,y=10)
name_lbl=CTkLabel(win,text="Emp_name:").place(x=10,y=50)
name_entry=CTkEntry(win,textvariable=v4,placeholder_text="Employee Name:").place(x=110,y=50)
loc_lbl=CTkLabel(win,text="Emp_Location:").place(x=10,y=90)
loc_entry=CTkEntry(win,textvariable=v2,placeholder_text="Employee Location:").place(x=110,y=90)
salary_lbl=CTkLabel(win,text="Emp_salary:").place(x=10,y=130)
sal_entry=CTkEntry(win,textvariable=v3,placeholder_text="Employee Salary:").place(x=110,y=130)
recr_btn=CTkButton(win,text="Insert",command=ins).place(x=10,y=180)
ret_btn=CTkButton(win,text="View Details",command=dele).place(x=170,y=180)
res=CTkTextbox(win,height=300,width=200)
res.place(x=50,y=210)
win.mainloop()
```

```
mysql> describe emp;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| e_id | int | NO | PRI | NULL | auto_increment |
| e_name | varchar(20) | NO | | NULL | |
| e_location | varchar(20) | NO | | NULL | |
| e_sal | int | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)
```

Output:



b. Design a database application to search the specified record from the database.

Code:

```

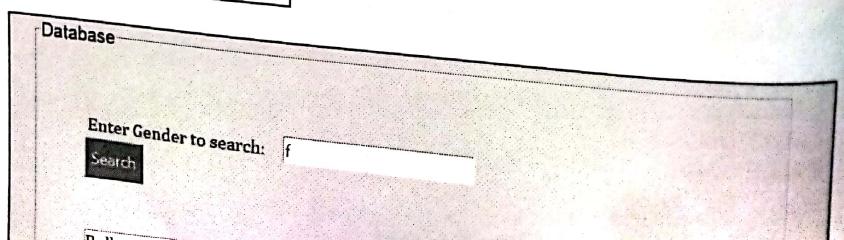
from tkinter import *
import mysql.connector
from tkinter import messagebox
db = mysql.connector.connect(host="localhost", user="root", password="123456",
                               database="syits")
def search():
    g = var1.get()
    cur = db.cursor()
    cur.execute("select * from stud where gender = %s", (g,))
    res = cur.fetchall()
    for x in res:
        resu.insert(END, f'Rollno: {x[0]}')
        resu.insert(END, f'First Name: {x[1]}')
        resu.insert(END, f'Last Name: {x[2]}')


window = Tk()
window.geometry("800x600")
var1 = StringVar()
f = LabelFrame(window, text="Database", height=500, width=800, font="aptos")
f.pack()
num1 = Label(f, text="Enter Gender to search: ", font="cambria")
num1.place(x=55, y=70)
ne1 = Entry(f, textvariable=var1)
ne1.place(x=280, y=70)
add = Button(f, text="Search", command=search, bg="grey", fg="white")
add.place(x=55, y=100)
resu = Listbox(f, font="cambria")
resu.place(x=55, y=200)
window.mainloop()
db.close()

```

Output:

	rollno	first_name	last_name	age	gender
▶	1	vedika	lalit	20	f
	2	sejal	kasale	19	f
	3	Athrva	Parab	20	m
	4	Ayush	Parab	20	m
	4	Ayush	Parab	20	m
	5	Madhura	Patil	21	m
	5	Shardul	Kudav	21	f



c. Design a database application to that allows the user to add, delete and modify the records.

Code:

```

import tkinter as tk
from tkinter import messagebox
import mysql.connector
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123456",
    database="syits"
)
def on_select(event):
    selected_index = listbox.curselection()[0]
    sel_pd, pname, qua = listbox.get(selected_index).split(' ')
    quan = ''.join([x for x in qua if x.isdigit()])
    clear_entries()
    name_entry.insert(tk.END, pname)
    quantity_entry.insert(tk.END, quan)
def clear_entries():
    name_entry.delete(0, tk.END)
    quantity_entry.delete(0, tk.END)

def update_listbox():
    listbox.delete(0, tk.END)
    c = conn.cursor()
    c.execute("SELECT * FROM prod")
    products = c.fetchall()

    for product in products:
        listbox.insert(tk.END, f'{product[0]} {product[1]} (Quantity:{product[2]})')

def add():
    pname = name_entry.get()
    quantity = quantity_entry.get()
    try:
        c = conn.cursor()
        c.execute("INSERT INTO prod (pname, quantity) VALUES (%s, %s)", (pname,
        quantity))
        conn.commit()
        update_listbox()
        clear_entries()
    except Exception as e:
        messagebox.showerror("Database Error", str(e))

def delete():
    try:
        c1 = conn.cursor()
        selected_index = listbox.curselection()[0]
        selected_product = listbox.get(selected_index).split(' ')[0]
        c1.execute("DELETE FROM prod WHERE id = %s", (selected_product,))
        conn.commit()
        update_listbox()
    except IndexError:
        messagebox.showwarning("Selection Error", "Please select a product to delete.")
    except Exception as e:
        messagebox.showerror("Database Error", str(e))

```

```

    messagebox.showerror("Database Error", str(e))

def modify():
    try:
        c2 = conn.cursor()
        selected_index = listbox.curselection()[0]
        sel_pid = listbox.get(selected_index).split(' ')[0]
        new_pname = name_entry.get()
        new_quantity = quantity_entry.get()
        c2.execute("UPDATE prod SET pname = %s, quantity = %s WHERE id = %s",
                   (new_pname, new_quantity, sel_pid))
        conn.commit()
        update_listbox()
        clear_entries()

    except IndexError:
        messagebox.showwarning("Selection Error", "Please select a product to modify.")
    except Exception as e:
        messagebox.showerror("Database Error", str(e))

root = tk.Tk()
root.title("Product Manager")

tk.Label(root, text="Product Name:").grid(row=0, column=0, padx=10, pady=10)
name_entry = tk.Entry(root, width=30)
name_entry.grid(row=0, column=1, padx=10, pady=10)

tk.Label(root, text="Quantity:").grid(row=1, column=0, padx=10, pady=10)
quantity_entry = tk.Entry(root, width=30)
quantity_entry.grid(row=1, column=1, padx=10, pady=10)

add_button = tk.Button(root, text="Add Product", command=add)
add_button.grid(row=2, column=0, padx=10, pady=10)

modify_button = tk.Button(root, text="Modify Product", command=modify)
modify_button.grid(row=2, column=1, padx=10, pady=10)

# delete_button = tk.Button(root, text="Delete Product", command=delete)
# delete_button.grid(row=2, column=2, padx=10, pady=10)

listbox = tk.Listbox(root, height=10, width=60)
listbox.grid(row=3, column=0, columnspan=3, padx=10, pady=10)

update_listbox()
listbox.bind('<>ListboxSelect>>', on_select)
root.mainloop()

conn.close()

```