

```
#fileobj = open("abc.txt", "w")      # file open
fileobj.write("Hello world\n")
fileobj.write(' I am Preet Jain In study in TCSC \n lives in kandivali')
fileobj.close()

fileobj = open("abc.txt", "r")      # read mode
#read
string1 = fileobj.read()
print("The output of read method = ", string1)
fileobj.close()

>>> ("The output of read method = ", 'Hello world\n I am Preet Jain\n in study in TCSC \n lives in kandivali\n')

# readlines()
fileobj = open("demo.txt", "r")
string2 = fileobj.readlines()
print("The output of readline method = "string2)
fileobj.close()

>>> ("The output of readline method = "Hello world \n")

# readlines()
fileobj = open("demo.txt", "r")
string3 = fileobj.readlines()
print("The output of readlines = "string3)
fileobj.close()

>>> ("The output of readlines = "Hello world\n I am Preet Jain\n in study in TCSC \n lives in kandivali\n)

# file attributes
string4 = fileobj.name
print("Name of file (name attribute) : "string4)
>>> ("Name of file (name attribute) : abc.txt")

string5 = fileobj.closed
print("close attribute = ", string5)

>>> ("close attribute = 'True'")
```

Aim: To demonstrate the use of different file accessing modes, different files, different and real methods

Step 1: Create a file object using open method and use the write access mode followed by writing some contents onto the files & then closing the file.

Step 2: Now, To open the file use the read mode by using the appropriate syntax. i.e. read(), readlines() and readline(), then store the output in variable & display the contents. of the Variable.

Step 3: Use the fileobject for finding the name of the file, the file mode which is opened whether the file is still open or close & finally the output of the softspace attribute.

PL

Step 4: Now open the file obj in write mode  
write some another content close  
subsequently. Then again open the fileobj  
in 'w+' mode that is the update mode  
and write contents.

Step 5: Open file obj in read mode display  
the update written content & close  
open again in 'rt' mode with  
parameter passed in display the  
output subsequently.

Step 6: Now open fileobject in append mode  
open write method write content  
close the fileobj again. Open the  
fileobject in read mode and display  
the append output.

step 7 : Open the fileobj in read mode, declare a variable and perform fileobj dot tell method and store the output consequently in variable.

step 8 : Use the seek method with the argument with opening the fileobj in read mode and closing subsequently.

step 9 : Open fileobj with read mode also use the readlines method and store the output consequently in and print the same for counting the length use the for condition statement and display the length.

Latif  
Jr. MCA

```
#append mode  
fileobj=open("abc.txt", "a")  
file obj.write("Python")  
fileobj.close()  
file obj=open("abc.txt", "a")  
str3=fileobj.read()  
Print("Output of append mode:", str3)  
fileobj.close()  
>>>('output of append mode:', 'Sweet', 'Python')
```

```
# tell()  
fileobj=open("abc.txt", "r")  
pos=fileobj.tell()  
Print("tell():", pos)  
fileobj.close()  
>>>('tell():' pos)
```

```
#seek  
fileobj=open("abc.txt", "r")  
str4=fileobj.seek(0,0)  
str8=fileobj.read(10)  
Print("The beginning of the file:", str8)
```

```
#finding length of diff lines exist within file,  
fileobj=open("abc.txt", "r")  
str9=fileobj.readlines()  
Print("Output:", str9)  
for line in str9:  
    Print(len(line))  
fileobj.close()  
>>>('Output: [college database]')
```

#file attributes

a = fileobj.name

print("name of file (name attribute): ", a)

>>>('name of file(name attribute)', 'abc.txt')

b = fileobj.closed

print("closed attribute: ", b)

>>>('closed attribute:', 'True')

c = fileobj.mode

print("file mode", c)

>>>('file mode', 'r')

d = fileobj.softspace

print("softspace", d)

>>>('softspace:', 0)

#w+ mode

fileobj = open("abc.txt", "w+")

fileobj.write("Preet")

fileobj.close()

#write mode

fileobj = open("abc.txt", "w")

fileobj.write("Preet")

fileobj.close()

#wt mode

fileobj = open("abc.txt", "wt")

str1 = fileobj.read(w)

print("obj output of wt", str1)

fileobj.close()

>>>('output of wt', 'Preet')

#readmode

fileobj = open("abc.txt", "r")

str2 = fileobj.read()

print('output of read mode', str2)

>>>('output of read mode',

Preet

## Program No. 2

Illustrate the use of `__iter__()` and `next()` method along with the class to display the odd number upto 100.

### Algorithms

Step 1: Define a class and within that define the `__iter__` method which will initialize the first element within the container object.

Step 2: For extracting the next element from the container with an arguments and compare the number of elements required in a container by using the conditional statement.

Step 3: Now create an object from the given class and pass this object as argument to the `iter()` method.

Step 4: Now using the conditional statement display all the values from the given container.

```

class odd:
    def __iter__(self):
        self.num = 1
        return self
    def __next__(self):
        if self.num <= 100:
            num = self.num
            self.num += 2
            return num
        else:
            raise StopIteration

```

```

myobj = odd()
myiter = iter(myobj)
for num in myiter:
    print(num)

```

Output:

1  
3  
5  
7  
9  
11  
13  
15

~~for all~~

Source code

Only:

```
fleobj = open("abctext", "w")
```

```
fleobj.write("Python is an programming language. In  
Python is an indented and interpreted  
language")
```

Except IOException:

```
    print("Program cannot run due to environmental  
error")
```

else:

```
    print("Operation has been successful")
```

Output:

>>>

Operation Successful

## Practical No. 3.

23

16/12/19

Aim: Programs to demonstrate exception handling

i) Programs to demonstrate the use of I/O error.

Step 1 :

Use the try block to define the normal course of action. For eg : Define the file object and open the file in the write or read mode and write context onto the file

Step 2 :

Use the except block with the I/O error as an environment error and convey the appropriate message to the user. ~~else~~ display the message that the operation is carried out successfully.

II) Program to demonstrate the multiple exception  
viz. IOError and ValueError.

Step 1:

Use the try block and define the file object and open the file in write or read mode and write some content onto the file.

Step 2:

Also accept the value from the user and if it is a valid value, display the entered value and terminate the condition by using the break statement.

Step 3:

Define the except blocks for IOError and ValueError.

Source Code:

```

while True:
    try:
        fileobj=open("abc.txt", "w")
        fileobj.write("Python is an indented language \n"
                      "Python is an interpreted language")
        a=int(input("Enter a number: "))
        print(a)
        break
    except IOError:
        print("There is an environment error")
    except ValueError:
        print("The given value is invalid.")

```

Output:

```

>>> Enter a number: Preet
The given value is invalid.
>>> Enter a number: 78
78

```

```
#Match()
import re
pattern="FYS"
sequence = "FYS represents computer science stream"
if re.match(pattern, sequence):
    print(`match pattern found!`)

else:
    print("Not Found!")
>>> print(matcher.pattern found!
```

#numerical values (segregation)

```
import re
pattern=r'\d+'
string = 'hello123.howdy 789, 45howdy'
output = re.findall(pattern, string)
print(output)
>>>[123, '789', '45']
```

```
#split()
import re
pattern= r'\d+'
string = 'hello123.howdy 789,45howdy'
output = re.split(pattern, string)
print(output)
>>>['hello', 'howdy', ' ', 'howdy']
```

## PRACTICAL No. 4

25

Topic : Regular expression

Step 1 : Import re module declare pattern and declare sequence use match method with declare arguments, if arguments matched then print the same otherwise print pattern  
Not Found!

Step 2 : Import re module/declare pattern with literal and meta character.  
Declare string value. Use the findall() with arguments and print the same.

Step 3 : Import re module/declare pattern with meta characters use the split() and print the output.

Program :-

```
from tkinter import *
def sel1():
    Selection = "Preet"
    Label.config(text=Selection)
def sel2():
    Selection = "TCSC"
    Label.config(text=Selection)
def sel3():
    Selection = "Kandivali"
    Label.config(text=Selection)
def sel4():
    Selection = "Thakur Village"
    Label.config(text=Selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any roll number")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1778", variable=var, value=0, command=sel1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="1779", variable=var, value=1, command=sel2)
R2 = Radiobutton(root, text="1780", variable=var, value=2, command=sel3)
R3 = Radiobutton(root, text="1781", variable=var, value=3, command=sel4)
R3.pack(anchor=N)
R2.pack(anchor=N)
R4 = Radiobutton(root, text="1781", variable=var, value=3, command=sel4)
R4.pack(anchor=N)
label = Label(root)
label.pack(side=BOTTOM)
label
root.mainloop()
```

30

OUTPUT

18	-	<input type="checkbox"/>	X
Select any roll number:			
01778			
01779			
01780			
01781			
Reset			

Aim: To make use of ScrollBar widget of the GUI application.

Algorithm:

Step 1: Import tkinter library to use scroll Bar widget

Step 2: Create an object corresponding to scroll parent window & create an object from scrollbar & place it on the parent window so created.

Step 3: Create an object of label method to provide a heading and place it on parent window.

Step 4: Use pack method along with object of scrollbar method & use argument side & fill.

Step 5: Create an object of listbox method and place it onto parent window with attribute yscroll command

Step 6: Use for loop to insert values in the object of list box by using insert method.

Step 7: Use config method along with scroll bar object use command attribute.

Step 8: Finally call the mainloop method.

Program:

from Tkinter import \*

root = Tk()

root.geometry('480x400')

l1 = Label(root, text="B Batch Roll number: ", bg="black", fg="white")

l1.pack()

scroll = Scrollbar(root)

scroll.pack(side=RIGHT, fill=Y)

mylist = Listbox(root, yscrollcommand=scroll.set, bg="light blue")

for num in range(41, 81):

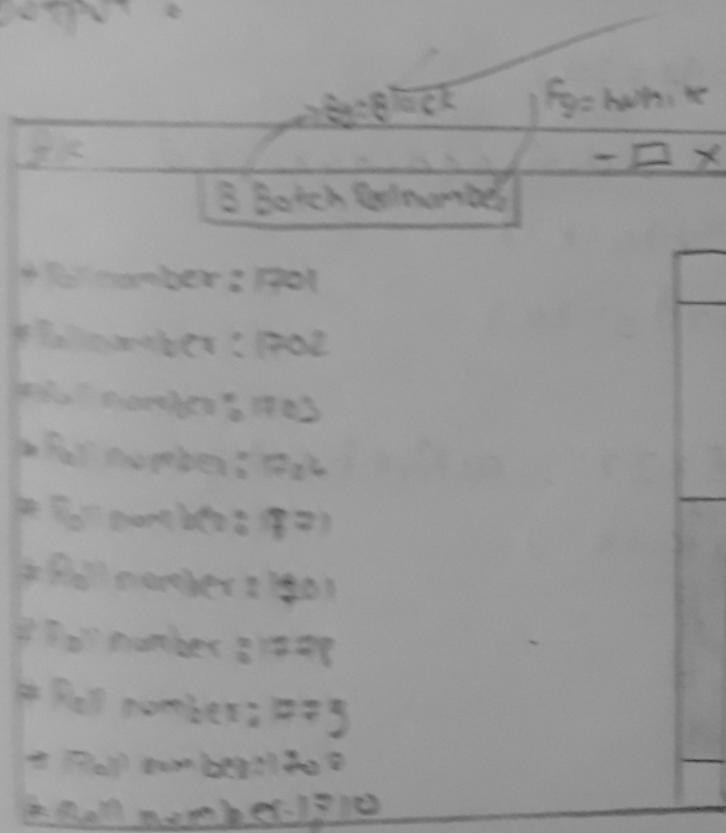
    mylist.insert(END, "# Roll number: " + str(num))

mylist.pack(side=LEFT, fill=BOTH)

scroll.config(command=mylist.yview)

root.mainloop()

Output:



Jyoti -

```
Program  
from tkinter import *  
window = Tk()  
window.geometry('100x200')  
frame = Frame(window)  
frame.pack()  
  
leftframe = Frame(window)  
leftframe.pack(side="LEFT")  
  
rightframe = Frame(window)  
rightframe.pack(side="RIGHT")  
  
b1 = Button(frame, text="Select", activebackground  
            ="Yellow", fg="black")  
b1.pack(side="LEFT", padx=10)  
  
b2 = Button(frame, text="Modify", activebackground  
            ="green", fg="black")  
b2.pack(side="RIGHT", padx=20)  
  
b3 = Button(frame, text="ADD", activebackground  
            ="pink", fg="black")  
b3.pack(side="BOTTOM", pady=20)  
  
b3.pack()  
  
b4 = Button(frame, text="EXIT", activebackground  
            ="red", fg="black")  
b4.pack(side="top")  
  
window.mainloop()
```

Aim: To make frame along with button widget.

Step 1: Import relevant methods from tkinter library.

Step 2: Object corresponding to parent window def the size of parent window in terms of no. of pixel.

Step 3: Now define the frame object from the method and place it onto the parent window.

Step 4: Create another frame object term as left frame and put it onto the parent window on it's left side.

Step 5: Similarly def the right frame and subsequently def the button object place onto the given frame with the attribute as text, active bg & fg.

Step 6: Now use the pack method along with side attribute.

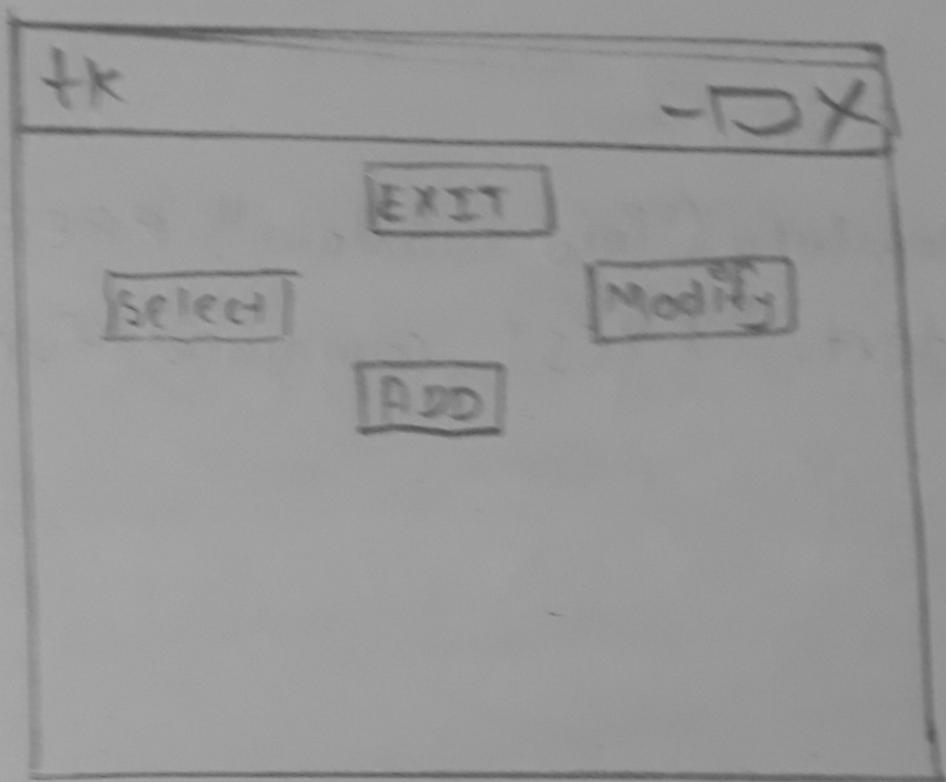
Step 7: Similarly create the button object corresponding to modify operation and put it into the frame object with side=RIGHT attribute.

Step 8: Create another button object and place it in right and label the button as 'Add'.

Step 9: Add another button and put it onto the right frame object and term it as 'EXIT'.

Step 10: Use the pack method simu. for all the objects and finally use the main loop method.

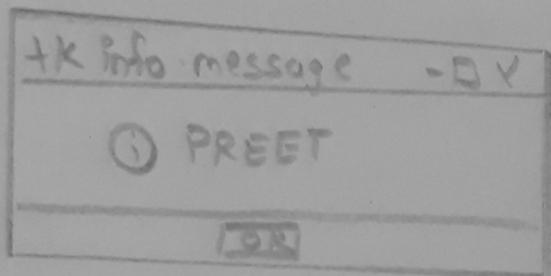
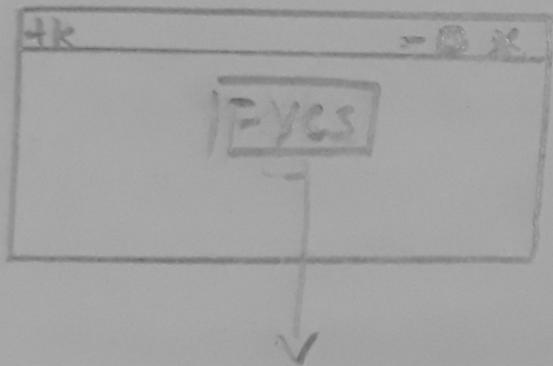
OUTPUT :-



Program :

```
from tkinter import *
from tkinter.messagebox
root = Tk()
def function():
    tkMessageBox.showinfo("info window", "PREET")
    b1 = Button(root, text="FYCS", command=function)
    b1.pack()
root.mainloop()
```

OUTPUT :



Aim: GUI component : message box

Step 1: Import the relevant method from tkinter library.

Step 2: Import tk Messagebox

Step 3: Define a parent window object along with the parent window.

Step 4: Define a function which will use tkmessage box with showinfo method along with the info window attribute.

Step 5: Declare a button with parent window object along with the command attribute.

Step 6: place the button widget onto the parent window and finally call mainloop() for triggering of the event called above

Jm1012

## PRACTICAL. 5cs)

Aim: To study different type of relief attribute

Step 1:

Import the relevant methods from the tkinter library along with parent window object declared.

Step 2:

Use the button object with the following attribute

- 1) Parent window
- 2) text attribute
- 3) relief

Step 3:

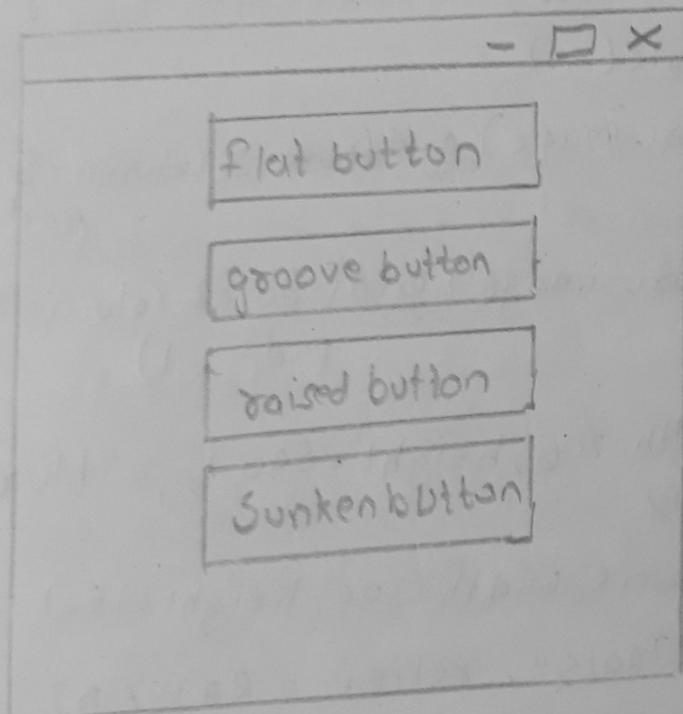
Use the corresponding pack method for the respective button object and trigger the corresponding event.

Step 4:

call the mainloop.

```
from tkinter import*
window=Tk()
b1=Button(window, text="Flat button", relief=FLAT)
b1.pack()
b2=Button(window, text="groove button", relief=GROOVE)
b2.pack()
b3=Button(window, text="raised button", relief=Raised)
b3.pack()
b4=Button(window, text="Sunken button", relief=SUNKEN)
b4.pack()
window.mainloop()
```

OUTPUT :



```

26
from tkinter import*
root=Tk()
root.title("Root")
root.minsize(200,300)
root.config(bg="red")
Frame=Frame(root)
Frame.pack()
leftframe=Frame(root, width=200, height=500, bg="green")
leftframe.pack(side=LEFT)
rightframe=Frame(root, width=200, height=500, bg="yellow")
rightframe.pack(side=RIGHT)
image=PhotoImage(file="C:\Users\KANCHAL\Desktop\toplt.gif")
ori_image=image.subsample(1,1)
label(leftframe,image=ori_image).grid(row=2, column=5, padx=1, pady=1)
label(rightframe,image=ori_image).grid(row=2, column=5, padx=1, pady=1)
upframe=Frame(root, width=200, height=500, bg="blue")
upframe.pack(side=TOP)
toolbar=Frame(upframe, width=200, height=150)
b1=Button(toolbar, text="Tools", relief=RAISED)
b1.pack(side=TOP, padx=2, pady=2)
b1=Button(toolbar, text="HELP", relief=RAISED)
b1.pack(side=TOP, padx=2, pady=2)
toolbar.pack(side=BOTTOM, padx=12, pady=10)
root.mainloop()

```

27

Aim: Write a program for displaying the image by using the concept of frame, tool bar, grid method & button method.

- Step 1: Import the relevant method from the tkinter library along with parent window object declared.
- Step 2: Create a object corresponding to parent window and use the following 3 methods - title, minsize, config.
- Step 3: Create a left frame object from the frame method and place it from the parent window with height, width and background attribute.
- Step 4: Now create a right frame object from the frame method with the width & the height specified and the row & column value should be specified.
- Step 5: Create a label object from the label method and place it onto the left frame. the txt attribute denoting the original image used as raised value and sub use the grid method with column value specified as 0,0 with

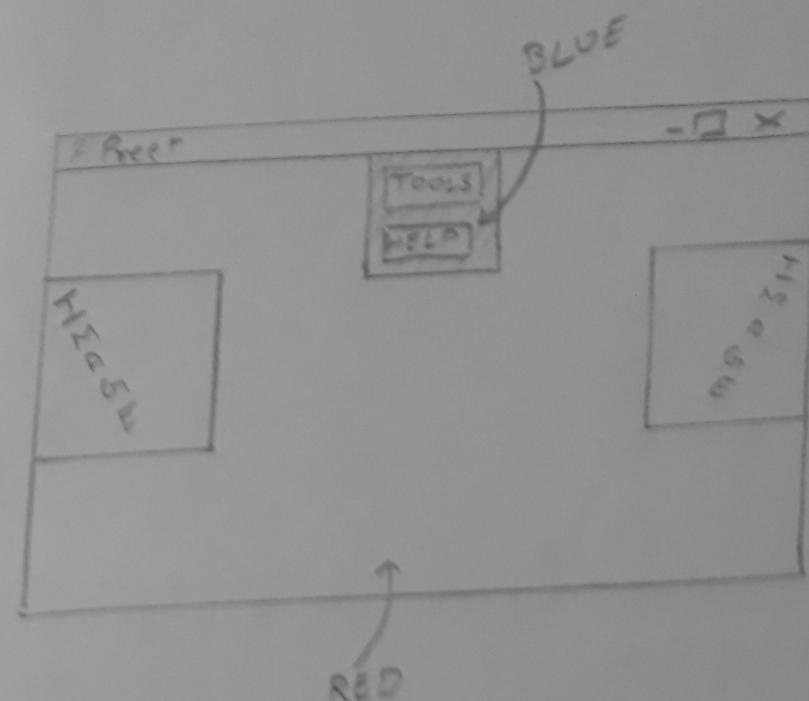
Step 1: Now use the photomerge method with the file attribute specified.

Step 2: Use the subImage method with the object of the image and give the x & y coordinate values

Step 3: Use the label method and position it onto the left frame & placing the image after the sampling & grid method for positioning in the first row.

Step 4: Create another label object position it onto the right frame and specifying the image & bg attribute @ 0.0

Step 5: Now create tool bar object from the frame method and position it on the topframe. By using button position the toolbar and close all the mailers



(1)

```
from tkinter import*
import random
questions=[ "How many keywords are there in C programming language?",  
"Which of the following function takes A console input in python?",  
"Which of the following is the capital of India?",  
"Which of the following is must to execute a python code?",  
"The Taj Mahal is located in?",  
"The append method adds value to the list at the?",  
"Which of the following is not a costal city of india?",  
"Which of the following is executed in browser(client side)?",  
"Which of the following keyword is used to create a function in python",  
"To declare a global variable in python we use the keyword?"]  
  
answers_choice=[  
["23","32","33","43"],  
["get()","input()","gets()","scanf()"],  
["Mumbai","Delhi","Chennai","Lucknow"],  
["TURBO C","Py Interpreter","Notepad","IDE"],  
["Patna","Delhi","Benaras","Agar"],  
["Custom location","end","center","beginning"],  
["Bengluru","Kochin","Mumbai","Vishakhapatnam"],  
["perl","css","python","java"],  
["function","void","fun","def"],  
["all","var","let","global"],  
  
]  
  
answer=[1,1,1,1,3,1,0,1,3,3]  
user_answer=[]  
indexes=[]  
  
def gen():
```

## PROJECT

39

Aim: Quiz game with the help of GUI widget.

Step 1: Import the relevant method from the tkinter library.

Step 2: Use the various attribute like title, geometry accordingly.

Step 3: Use the image method along with the label to place the image on parent window.

Step 4: Now use the label method to place your name and use various attribute like font, bg, fg.

Step 5: Use the Button widget to start the quiz and give the appropriate command.  
says: click

Step 6: Again use the label method to show the rules of the quiz.

Step 7: Now call the command(s) by defining it

Step 8: Now in that destroy all the previous attribute by using the destroy function to clear the screen

to start the quiz def another

step 10: Now to display the question use another label method along with that use the varios attribute to beautify the code like font width, justify, wraplength, background, padding and many more.

step 11: Now for displaying the radiobutton use the radio variable as int var and set the default value as (-1) so that by default the radiobutton will not be checked

step 12: Now use the r1, r2, r3, r4 for radiobutton and the the value 0, 1, 2, 3 respectively along with that use the intvar and command respectively

step 13: Now for question and the answers\_choice you can take the help of the array within that array give your questions and choices which has to be displayed to the user accordingly in both arrays.

```
x=random.randint(0,9)    ↗(2)
if x in indexes:
    continue
else:
    indexes.append(x)
def showresult(score):
    lblQuestion.destroy()
    r1.destroy()
    r2.destroy()
    r3.destroy()
    r4.destroy()
    labelfr=Label(root,text="you have scored {} marks out of 25".format(score),
    font=("GOUDYSTO",56),background="red",foreground="white")
    labelfr.pack(pady=20)
    labelt=Label(root,text="THANK YOU FOR PLAYING THE QUIZ:",anchor="center",font=("GOUDYSTO",56),background="black",foreground="green")
    labelt.pack(pady=50)
    labelf1=Label(root,text="DO COME AGAIN :)",font=("GOUDYSTO",56),background="blue",foreground="white")
    labelf1.pack(pady=100)
def calc():
    global indexes,user_answer,answer
    x=0
    score=0
    for i in indexes:
        if user_answer[x]==answer[i]:
            score=score+5
        x+=1
    print(score)
    showresult(score)
```

```

def selected():
    global radiovar, user_answer
    global lblQuestion, r1, r2, r3, r4
    global ques
    x=radiovar.get()
    user_answer.append(x)
    radiovar.set(-1)
    if ques<5:
        lblQuestion.config(text=questions[indexes[ques]])
        r1['text']=answers_choice[indexes[ques]][0]
        r2['text']=answers_choice[indexes[ques]][1]
        r3['text']=answers_choice[indexes[ques]][2]
        r4['text']=answers_choice[indexes[ques]][3]
        ques=ques+1
    else:
        print(indexes)
        print(user_answer)
        calc()
def startquiz():
    global lblQuestion, r1, r2, r3, r4
    lblQuestion=Label(root, text=questions[indexes[0]], font=("Bodoni MT", 36), width=500, justify="center", wraplength=400, background="white")
    lblQuestion.pack(pady=(100, 30))
    global radiovar
    radiovar=IntVar()
    radiovar.set(-1)

```

(3)

Step 14: Now call the given function which you had use in radio button.

11

Step 15: Now to store the answers which is get selected by the user declare the variable with that use the radiovar along with the get which will display the get store in the array i.e user\_answer (an empty array to store the answers of the user). along with that use append()

Step 16: Now create an empty array name it as Pndexes for storing the question which will be displayed to the user

Step 17: Now we want to display the the 5 question out of 10 call the another function as gen() in that by using the while loop show that the loop will be terminate till the len of Pndexes will be equal to 5.

Step 18: Now within the selected function use the if condition such that with the click of the radiobutton the question and the option has to be change. for question and radiobutton use the config with text such that it will change with sel

1:

step 19: Now call the function which will calculate the score give the initial value and 0.0 as start and x resp. Use the if condition if the user.answers[i] will equal to answers[i] then the score will increase by 5 and x will increase.

step 20: Now call the another function which will display the marks. Before that use the destroy function to clear the screen all the various label method for displaying the score and saying congrat and use the padding for

step 21: Call the mainloop.

-X-X-

def calculate\_score(score=0.0, user\_answers=[], answers=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]):  
 if user\_answers == answers:  
 score += 5  
 return score

def display\_marks(score):  
 print("Score: " + str(score))

def congrat():  
 print("Congratulation")

def mainloop():  
 while True:  
 user\_answers = [int(input("Enter answer " + str(i) + ": ")) for i in range(10)]

score = calculate\_score(score=0.0, user\_answers=user\_answers, answers=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

display\_marks(score)

if user\_answers == answers:  
 congrat()

else:  
 print("Incorrect answer")

if \_\_name\_\_ == "\_\_main\_\_":  
 mainloop()

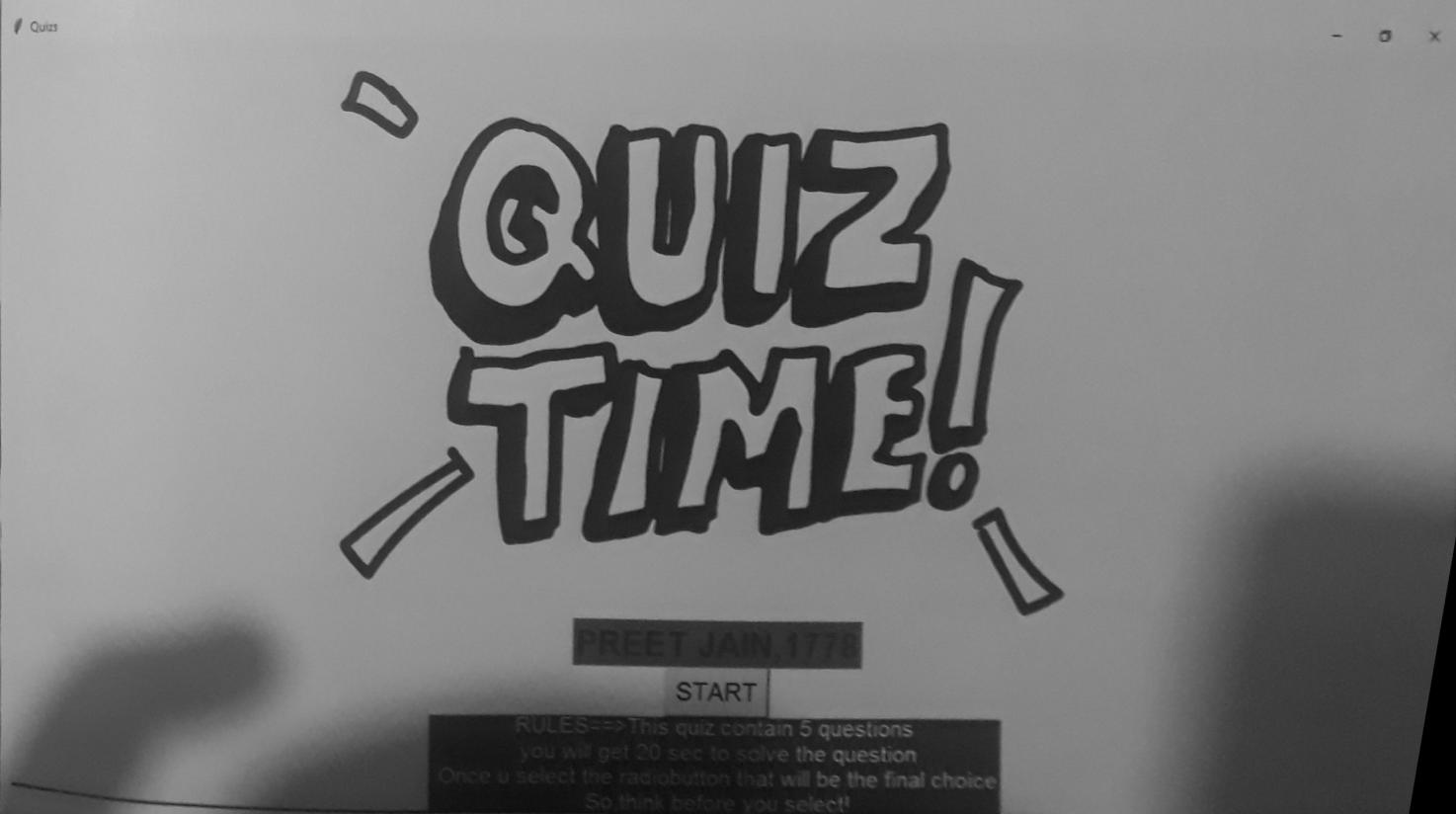
```


l1=Label(root,image=img)
l1.pack(padx=0)
labeltext=Label(root,text="PREET JAIN,1778",font=("Algerian Regular",24,"bold"),background="brown",foreground="black")
labeltext.pack()
b1=Button(root,text="START",relief="raised",command=startispressed,background="yellow",foreground="black",font=('None',18))
b1.pack()
instruction=Label(root,text="RULES==>This quiz contain 5 questions\n you will get 20 sec to solve the question\n Once u select the radiobutton that will be the final choice\n So,think before you select!",background="purple",foreground="white",font=('LCALLIG',16))
instruction.pack()
root.mainloop()

```

## OUTPUT:

Slide 1



Slide 2:

The append  
method adds value  
to the list at the?

- A. append
- B. add
- C. remove
- D. join

Slide 4:

The Taj Mahal is  
located in?

- A. New Delhi
- B. Mumbai
- C. Bangalore
- D. Lucknow

Slide 6:

How many  
keywords are there  
in C programming  
language?

- A. 30
- B. 32
- C. 33
- D. 42

slide 3:

Which of the  
following is  
executed in  
browser(client  
side)?

- A. perl
- B. c++
- C. javascript
- D. java

slide 5:

Which of the  
following is the  
capital of India?

- A. Mumbai
- B. Delhi
- C. Chennai
- D. Lucknow

41

Slide 7:

you have scored 20 marks out of 25

THANK YOU FOR PLAYING THE QUIZ

DO COME AGAIN :)

-X—X—X-

## Practical - 6

Aim: To make use of various of memory

Step 1: Import db library and use the open method for creating the data base by specifying name at the data base along with the corresponding flag.

Step 2: Use the objects for accessing to given size and the corresponding region for the web size.

Step 3: Check whether the given URL address with the regular of the pages is not equal to none than display the message from the URL address else NOT found.

```
import db  
db = dbm.open("data base", flag="c")  
if db["www"] != None:  
    print("good")  
else:  
    print("Not good")  
db.close()
```

Output:

Good

Aim: To make use of various types of database statements.

Step 1: Import the corresponding library taking data base connection.

Step 2: Now create Connection Objects using sq library and connecting method for creating the new database.

Step 3: Now create the Cursor Objects using cursor method from the Connection object created in steps

Step 4: Now use the executing method for creating the table with the column name and respective data type.

Step 5: Now with the Cursor objects use insert statements for entering the values co-ordinate into the different field considering data type.

Step 6: Use the Commit method to complete the transaction. Use the Connection object

Step 7: Use the execute statement along with the object for executing the value the db using select

Step 8: Finally use the fetchall method to

Programs  
import os,sqlite3  
connection = sqlite3.connect("student.db")  
c1 = connection.cursor()  
c1.execute('Create table student(Name, RNO,DOB)')  
c1.execute('Insert into student values("Rakesh",  
1840, 23-06-2002)')  
c1.execute('Insert into student values("Sachin",  
1841, 25-04-1996)')  
c1.execute('Insert into student values("Manoj", 1842,  
02-02-2002)')

46

connection.commit()  
c1.execute('Select \* from student')  
c1.fetchall()  
c1.execute('Drop table student')

Output:

[('Rakesh', 1840, 23-06-2002), ('Sachin', 1841, 25-04-1996),  
('Manoj', 1842, 02-02-2002)]

e.g.: Use the execute method and the drop table syntax  
for terminating the database finally use the close  
method