

```

180
fileobj=open("abc.txt","w") #file.open('write mode')
fileobj.write("Computer Sc Subject" + "\n")
fileobj.write("DAMS \n python \n DS \n")#file
fileobj.close() #closing file.

181
fileobj=open("abc.txt","r") #read mode
str1=fileobj.read() #read()
print("The output of read method: ",str1)
fileobj.close()

182
str2=fileobj.readline() #readline()
print("The output of readline method: ",str2)
str2=fileobj.readline() #readline()
print("The output of readline : ",str3)

>>>The output of read method:Computer Sc Subject
None
Python
DS
>>>The output of readline method: Computer
Science Subject
>>>(The output of readline or: ['Computer',
Science Subject\n', 'DAMS\n', 'Python\n', 'DS\n'])

```

Practical 02

02

* Objective: Demonstrate the use of different file accessing mode, different attributes.

Step 01: Create a file object using open method and use the write() access mode followed by writing some content onto the file and then closing the file.

Step 02: Now open the file in read mode and then used read(), readline() methods and store output in variable.

Step 03: Now use the functions for finding the name of the file, the file mode in which it is opened whether the file is still open or close and finally output of tell() attribute.

Q3

Step 04: Now open the file in write and write some more content like already, then again open the file in 'w+' mode, that is the update mode and write contents.

Step 05: Open file object in read mode, display the update written contents and then open update written contents and clear them again in 'r+' mode with presentation cleared and display the output subsequently.

Step 06: Now open file object in append mode open with write method and close the file object again. Open the file object in read mode and display the append mode.



Q4

a. Fileobj.name

```
print("Name of file (name attribute)", a)
```

and Name of file (name attribute), 'abc.txt'

b. Fileobj.read

```
print ("File's attribute", b)
```

```
>>> ('(File's) attribute', True)
```

c. Fileobj.mode

```
print (Fileobj.mode', c)
```

```
>>> ('Fileobj.mode', 'r')
```

d. Fileobj.readline

```
print ("Fileobj.readline", d)
```

```
>>> whitespace: 0
```

e. File mode

```
fileobj = open('abc.txt', 'wt')
```

```
fileobj.write ("Bansuri Banay")
```

```
fileobj.close()
```

f. r+ mode

```
fileobj = open('abc.txt', 'r+')
```

```
ctrl = fileobj.read (r)
```

```
print ("Output of r+", ctrl)
```

```
fileobj.close ()
```

```
>>> ('Output of r+', 'Bansuri')
```

180

Step 07: Open the file in read mode, declare a variable and perform `fileobj.read()` dot tell method and store the output consequently in variable.

Appended mode

```

fileobj = open("abc.txt", "a")
fileobj.write("data structures")
fileobj.close()
fileobj = open("abc.txt", "r")
str = fileobj.read()
print("Appended mode : ", str)
fileobj.close()
# Now open a append mode, 'sharly', 'data structure'
fileobj = open("abc.txt", "a")
fileobj.write("data structure")
fileobj.close()
fileobj = open("abc.txt", "r")
pos = fileobj.tell()
print("tell : ", pos)
fileobj.seek(0)
fileobj.close()
fileobj = open("abc.txt", "r")
str = fileobj.readline()
str1 = fileobj.read(10)
print("The beginning of the file : ", str)
print("The length of the file : ", str1)
fileobj.close()

```

181

Step 08: Use the `seek()` method with the argument with opening the fileobj in read mode and doing subsequently.

Step 09: Open `fileobj` with read mode after use `tell()` readline method and store the output consequently in and print the same so counting the length use the for condition and display the length.

Printing length of different lines with while loop

```

fileobj = open("abc.txt", "r")
str = fileobj.readline()
print("Length : ", len(str))
for line in str:
    print(len(line))
    fileobj.readline()
>>> College Database I

```

Soham
Date: 20/07/2021

200 Practical 2

Ques - To display elements of a tuple using
another method.

Algorithm:

Step 1: Turn a tuple with certain elements
named in it.

Step 2: Use iter method with tuple and size
it to a variable.

Step 3: Use the next method with variable
and print the elements.

Ques - To use iter method with for loop
Algorithm:

Step 1: Turn a tuple with certain elements
named in it.

Step 2: Use the for (conditional statement) to
iterate each element of tuple.

Step 3: Print the elements of tuple.

* Program:

```
mytuple = ("Ranay", "Neeraj", "Sachin", "Raj")
myiter = iter(mytuple)
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

& Output:

```
Ranay
Neeraj
Sachin
Raj
```

* Program:

```
mytuple = ("Ranay", "Neeraj", "Sachin", "Raj")
for a in mytuple:
    print(a)
```

& Output:

```
Ranay
Neeraj
Sachin
Raj
```

1/1/2020

190

Program:

class odd:
def __init__(self):

self.even = 0
return self

def next(self):

return self.even
self.even += 2
return even

myobj = odd()

oddnumbers = myobj()
for i in range(10):

if i < 2:
print(i)

Output:

Enter / number: 6

1

3

5

191

* Aim : To find odd numbers in given range.
using function.

* Algorithm:

Step 1: Define a class which will contain
various func.

Step 2: Define the func method with an
argument and return the value of
arrangement.

Step 3: Define a func which increments the
value of argument by two.

Step 4: Create an object which inherits the
~~properties~~ of class and take the
user input.

Step 5: Use the for conditional statement
followed by if conditional statement
and print the answer.

Ques

- * write a program to print first 10 numbers using class.
- * signature:
 - step 1: take a class which will contain values from 1 to 10.
 - step 2: take a function which will return the value of argument and return the value of argument.
 - step 3: take and modify which increases the value of argument by 1 and prints it.
 - step 4: Create a object which inherits the property class and take the value right.
 - step 5: See for loop to print the value of the variable.

* Program:

```
class myclass:
    def __init__(self):
        self.a = 1
    def fun(self):
        if self.a <= 10:
            x = self.a
            self.a += 1
            return x
        else:
            return "stop"
myobj = myclass()
myfun = myobj.fun()
for x in myfun:
    print(x)
```

* Output:

180

```
# Sequence  
list = [1, 2, 3, 4, 5]  
empty = []  
for i in list:  
    empty.append(i * 2)  
print(empty)
```

Output:

[2, 4, 6, 8, 10]

180

* Aim: To find square of a number without using multiplication.

* Algorithm:

Step 1: Define a list, which contains certain values.

Step 2: Define an empty list.

Step 3: Use for loop followed by append method to append the output into the empty list.

Step 4: Print the value of the list.

Practical No. 3

a. write a program using the except block related to the KeyboardInterrupt error

step 1: On the try use to define the normal kind of action for ex. Define the class and write the file into the model and write some content into the file.

Step 2: Use the except block with the KeyboardInterrupt error to catch the error and if caught the appropriate message to the user else display that the operation is carried out successfully.

Program:

try:

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

fileobj.write("Hi we are trying")

except KeyboardInterrupt:

print("This is to Error")

else:

print("This operation succeeded.")

Q1:

```
with open("text.txt", "w") as f:
    f.write("Hello World")
    f.close()
    print("File has been written")
```

Q2:

```
try:
    value = int(input("Enter a number: "))
except ValueError:
    print("This is not a number")
else:
    print("Input was successful")
```

Q3:

Write a program to demonstrate the use of value error in the given program statement.

Step 1: Accept the value from the user if it is a valid value display the entered value & terminate the control by using break statement

Step 2: If accept the value from the user and if it is a valid value display the entered value and terminate the control by using the break statement.

Step 3: When we except ValueError for Integer and ValueError

DW

Practical - 4

RegEx

```
match()
import re
pattern = "for loop"
string = "for loop"
if re.match(pattern, string):
    print("Matched pattern found!")
else:
    print("NOT found!")
```

matched pattern found!

Unmatched pattern found (ognition).

```
import re
pattern = "d+"
string = "Hello123"
if re.match(pattern, string):
    print("Matched pattern, string")
else:
    print("NOT found")
```

#Matched

```
import re
pattern = "d{3}"
string = "Hello123"
if re.match(pattern, string):
    print("Matched pattern, string")
else:
    print("NOT found")
```

```
③ re.split():
import re
String = "abc def ghi"
pattern = " "
replace = ""
if re.split(pattern, String):
    print("abc def ghi")
```

Practical - 4

RegEx - 2

Step 1: Import re module define pattern and declare variable 'abc' with match method with 'abc' arguments & arguments matched then print the value otherwise print pattern NOT FOUND!

Step 2: Import re module define pattern with literal and meta characters declare string value via the input() with arguments and print the same.

Step 3: Import re module define pattern with meta characters via the split() and print the output

Step 4: Import re module define string and accordingly define pattern via via the back space with no-space via the split() with 3 arguments and print the string without spaces

Practical - 4 Regular Expressions

```
100
match()
import re
pattern = "XYZ"
string = "This represents computer science"
output = re.match(pattern, string)
if output != None:
    print("Matched pattern found!")
else:
    print("Not found!")
#nonmatch()
print("Nonmatch pattern found (exception).")
import re
pattern = "[abc]"
string = "Hello123", "muddy123", "4schau"
output = re.search(pattern, string)
print(output)
#split()
print("Hello", "Howdy", "Hooray")
print("Hello", "Howdy", "Hooray")
import re
pattern = "[abc]"
string = "Hello123", "muddy123", "4schau"
output = re.split(pattern, string)
print(output)
#replace()
import re
string = "abc def ghi"
pattern = "abc"
replace = "xyz"
output = re.sub(pattern, replace, string)
print(output)
```

Practical - 5

TOPIC: Regular expression & Applications

Step 1: Import re module define pattern and declare regular expression object matched with different arguments & segments matched then print the other otherwise print pattern NOT FOUND!

Step 2: Import re module define pattern with literal and meta characters. Declare string value use the findall() with arguments and print the same.

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

Step 4: Import re module define string and accordingly define pattern replace the blank space with no-space use sub() with 3 arguments and print the string without spaces.

Program 1

Step 1: Import re module, declare a regular expression method for finding digits in the given string. Use the search() function with dot product as search() gives matching pattern as group(). It will also return the matched string.

Q) If group() is called
Import re module. Python is an interesting language.
Explained. Python is an interesting language.
Research ("PythonExplained")
print(**group**)

Ans: print(**group**)
=> <_sre.SRE_Match object at 0x0000000004000000>
Pattern:

Q) Program to match the given set of phone numbers:
Import re
set1=[("9898999000", "9999999999")]

for value in set1:
 if re.match(r"9[0-9]{9}", value):
 print("Matched")

print("Criteria matched for all phones")

else:
 print("Criteria Failed.")

=> Criteria matched for all phone
Criteria matched for all phone.

Program 2

Step 1: Import re module, declare list with numbers. Use the conditional statement. Here we used the for conditional statement. Use if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered numbers are equal to 10.

Step 2: Import re module, declare a string. Use the module with findall() for finding the result in the string.

Step 3: Import re module, enter a string, use pattern to display only two digits of the particular string. Use findall() before this variable with which we are zero use the for condition and subsequently use the if condition to check whether condition satisfies add up or the incorrect value and display the value is subsequently.

Q) Program
if input is male, declare a boy
and print name in the bracket () and
print the output respectively.

Q) ^{WAP} To calculate the number of males and females
input is
e.g. M.Ram,M.Ram,M.Ram,M.Ram
P: i[1 to 4] ;
else: female(p);
end(0)
end(0)
P: 0
for i in C
if (M == 'M') :
 f=f+1
else:
 f=f-1
print ("No. of males is: ", f)
print ("No. of females is: ", f)
else [M1, 'M2', 'M3', 'M4']:
 No. of males is: 3
 No. of females is: 2
Dw

Practical 6
 Date: 10/08/2018
 Name: "my name is Harry."
 T1 pack (color="red", grade=20, party=20, quads=40)
 T1 pack (color="blue", grade=20, party=20, quads=40)
 T1 pack (color="green", grade=20, party=20, quads=40)
 T1 pack (color="yellow", grade=20, party=20, quads=40)

a) Output

```

graph TD
    W1["my name is Harry."]
    W2["color:red  
grade:20  
party:20"]
    W3["color:red  
grade:20  
party:20"]
    W1 --- W2
    W2 --- W3
  
```

b) Questions

Step 1: To make use of our original string code, the basic pack method

Step 2: Use the JButton library for importing the buttons of the widget

Step 3: Create a variable from a text variable to position it onto the parent window

Step 4: Use the pack() along with the object created from text method & use the frame.pack()

Step 5: Use the main loop method for triggering corresponding event

Step 6: Now repeat above step with a label method which takes the following argument
 i) Name of parent window
 ii) Text attribute which defines the string
 iii) The background color (bg)
 iv) The foreground color (fg)
 Thus used pack() with relevant attributes

Ques

a) How to make use of Indirective indirect function or multiple use of the option.

b) Aggregation:

Step 1: Use the factory method to inherit the relevant method.

Step 2: Define a "final" which tells when along given selection made from multiple options available.

Step 3: Use the factory method along with its method & call the variables as arguments within method.

Step 4: Now define the parent window to define using control variable.

Step 5: i) Partitioning on parent window.

ii) Text variable.

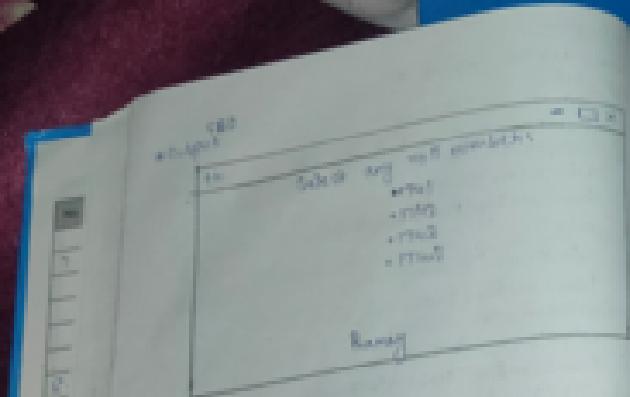
iii) Define variable argument.

iv) Comparing value and trigger the given function.

Step 6: Now define a label object and place in onto parent using pack method.
A. Finally use main loop method.

* Program

```
from tkinter import *
def main():
    selection = "None"
    Label.config(text=selection)
def radio1():
    selection = "Top"
    Label.config(text=selection)
def radio2():
    selection = "Bottom"
    Label.config(text=selection)
root = Tk()
var = IntVar()
l1 = Radiobutton(text="Select any one number")
l1.pack(side="TOP")
l1 = Radiobutton(text="Top", variable=var, value=1,
                 command=radio1)
l1.pack(side="TOP")
l1 = Radiobutton(text="Bottom", variable=var, value=2,
                 command=radio2)
l1.pack(side="TOP")
l1 = Radiobutton(text="None", variable=var, value=3,
                 command=main)
l1.pack(side="TOP")
Label = Label(text="None")
Label.pack(side="TOP")
```



- Ques: To make use of scroll bar widget of the GUI application.
- * Algorithm:
- Step 1: Import `JScrollPane` to use scroll bar widget.
 - Step 2: Create an object corresponding to scroll pane & window & create an object from `JScrollPane` & place it on the scroll pane we just created.
 - Step 3: Create an object of `JLabel` method to provide a heading and place it on parent window.
 - Step 4: Use `pack` method along with object of scroll bar method to use against component size.
 - Step 5: Create an object of `JButton` method and place it onto parent window with desired year.
 - Step 6: Use `for loop` to loop value in the object of `JLabel` by using `setText` method.
 - Step 7: Finally call `mainloop` method.

1

— Output

200

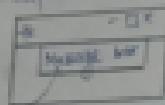
	Customer Name	Phone Number
1.	John Doe	(555) 123-1234
2.	Jane Doe	(555) 123-1235
3.	Bob Smith	(555) 123-1236
4.	Susan Smith	(555) 123-1237
5.	Mike Johnson	(555) 123-1238
6.	Anna Johnson	(555) 123-1239

— 1 —

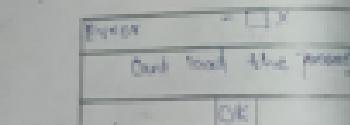
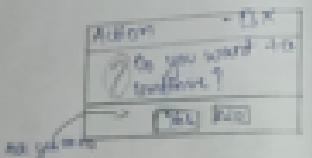
(a) Import from tkinter import *
from tkinter import messagebox
def message_box(Action): "Do you
want to cancel?"
message_box("Cancel", "Can't cancel
the process")
not (b)
returning (by "grey")
buttonbox ("ok", "cancel", "grey box", by "blue"
"cancel", "cancel").

o pack()
root.mainloop()

o output:



click on the
button.



Focus
Messagebox

Practical - 6

841

a aim To make use of messenger method one
of the user appeal.

b algorithm:

Step 1: Import relevant module from tkinter library.

Step 2: Define a function and use messagebox along
with different methods available
which contains one or more
arguments.

Step 3: Create an object from button method
and place it into the parent
window with text and command
attribute specified.

Step 4: Use pack method and finally use
the mainloop method.

140

- a) Write program to handle various windows using Tkinter button widget.
- b) Algorithm:
 - Step 1: Define a "root" and create a object by using `Tk()`, like `root = Tk()`.
 - Step 2: Define a button object which will be placed on the current window to handle and define another button which will be used to exit from the window and place it onto current window.
 - Step 3: Define another field which will use the `quit` method.
 - Step 4: Now create an object of main window and its config, like generating its.
 - Step 5: Define two button which will be placed on the main window for trapping and one to terminating.
 - Step 6: Finally call the `mainloop` method.

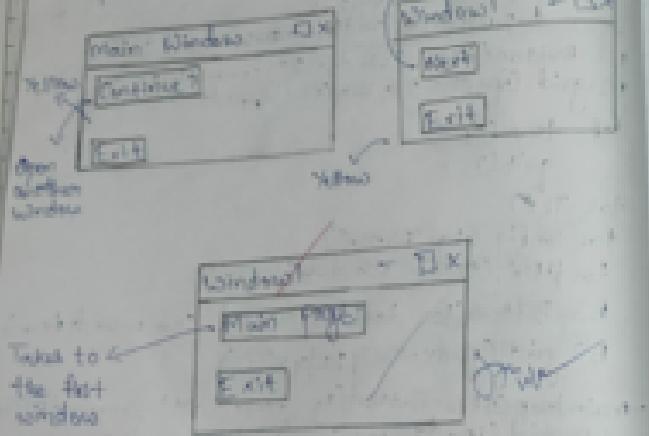
c) Program:

```
from Tkinter import *  
import time  
def main():  
    root = Tk()  
    root.title("My Window")  
    root.geometry("400x400")  
    root.minsize(400, 400)  
    B1 = Button(root, "window1", command=main)  
    B1.grid(ipadx=20, ipady=20, padx=20, pady=20)  
    def make2():  
        top = Toplevel()  
        top.title("Form 2")  
        top.minsize(400, 400)  
        B2 = Button(top, "window2", command=make2)  
        B2.grid(ipadx=20, ipady=20, padx=20, pady=20)  
        def form2():  
            quit()  
        B2.grid(ipadx=20, ipady=20, padx=20, pady=20)  
        B3 = Button(top, "Exit", command=top.destroy)  
        B3.grid(ipadx=20, ipady=20)  
    B1.grid(ipadx=20, ipady=20, padx=20, pady=20)
```

```

    def main():
        top=Tk()
        top.title("Window")
        top.config(bg="yellow")
        mainloop()
        a.grid(padx=20, pady=20)
        a.grid(padx=20, pady=20, columnspan=2, rowspan=2)
        a.grid(padx=20, pady=20)
        a.grid(padx=20, pady=20)
        a.mainloop()
    
```

Output:-



Practical :-

Ques

Step 1:

- > Create an object from the class and create an object from the object.

Step 2:

- > Use the .pack method to provide the direction very easier method.

Step 3:

- > Use the mainloop method to terminate.

29

a) Grid Window

- > Step 1:
 - > Create an object from grid window and put the grid method with the attribute get the grid.

- > Step 2:
 - > Create an object from the label method and put it into the grid window with the grid method and use the text attribute and use the grid attribute and use the row and column.

- > Step 3:
 - > Similarly create a second grid window object and add onto the grid window with orientation specified.

- > Step 4:
 - > Now use the mainloop method to display.

30

From Tkinter import *

root = Tk()

l1 = Label(root, text="Please enter 1-100 by 100")

l1.pack(side="left")

e1 = Entry(root, width=10, font="10, by 100")

e1.pack(side="top")

protoloop(1)

def protoloop(i):

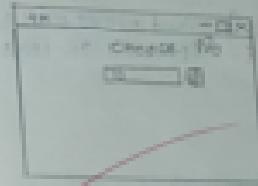
l2 = Label(root, text="Please enter 1-100 by 100")

l2.pack(side="left")

e2 = Entry(root, width=10, font="10, by 100")

e2.pack(side="top")

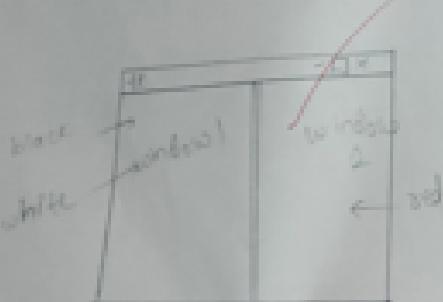
protoloop(i+1)



```

a JFrame
    .setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
    .add(new JPanel() {
        .setPreferredSize(new Dimension(300, 200))
        .add(new JLabel("Hello World", "green"))
    })
    .pack()
    .setVisible(true)

```



Canvas Widget:

Step 1:

Use the `create` method and create an object from the `Canvas` method and set its attribute height, width together with the parent window object.

Step 2:

Use the `create` method create and create two and create one along with the `Canvas` object so created and via the `content` value like use the `fill` attribute to assign various colors.

Step 3:

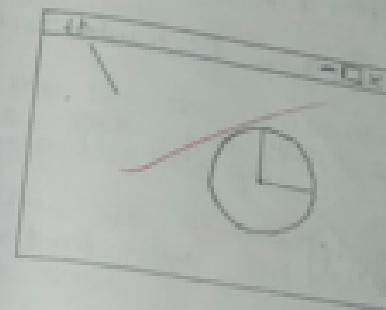
Now call the `frame` method and `mainloop` methods.

Java

→ Square (edge):

```
from turtle import *
t = turtle()
cl = canvas.Canvas()
height = 400, width = 400, bg = "white"
oval = cl.create_oval(20, 240, 180, 280, fill = "blue")
arc = cl.create_arc(20, 240, 180, 280, start = 0, extent = 180)
line = cl.create_line(10, 100, 100, 100, fill = "green")
cl.getTk().mainloop()
```

Output is



Dr min

© Microsoft Corporation

Project 3: **Friends**
Implementation: Create class ("Friends")
Create attribute ("Create table Friends (Name, Age, City, Sex, Job)");
insert attribute ("Create table Friends");
update attribute ("Update Friends values ('Homer', 17, 'Male', 'Teacher')");
delete attribute ("Delete from Friends where Name = 'Homer'");
insert attribute ("Insert into Friends values ('Bart', 12, 'Male', 'Student')");
create attribute ("Create table Friends values ('Maggie', 1, 'Female', 'Nurse')");
exception convert();
insert attribute ("Select name from Friends values ('Homer')");
select attribute();
select attribute ("Select name from Friends values following
select attribute());

第 9 章

~~[('Mahab'), ('Dived')]~~

14

11

847

Practical 8.
Sonic-
card
wave
harmonic
and
tubular
harmonics.

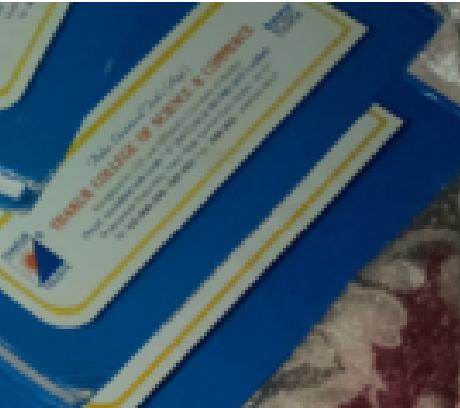
Digitized by srujanika@gmail.com

- ④ Import the corresponding money table of data base connection.
 - ⑤ Now Create Connection object using生活的 the new database.
 - ⑥ Now Create the insert object using cursor method.
 - ⑦ Now use the insert method for inserting the task with the column name and respective data type.
 - ⑧ Now with the cursor split using cursor method from the connection object created in step
 - ⑨ Use commit method to complete the transaction between created objects.

180

- ① Use the create statement along with the insert syntax for specifying the value the data base using table from where clause.
- ② Finally we use the drop table command for dropping the table for the table using certain object.
- ③ Use the truncate method and the drop table syntax for truncating the database. Finally we use the drop method.

181



Java Code:

```
import java  
db; dbm open("data base") "file" "c"  
if db["Name"] != Name  
    print("good")  
else:  
    print("Not good")
```

Output:
good

Q1

Q1 To make use of Database library.
Algorithm:
1) Import DB library and use the open
method for creating the data base
by specifying name of the data base
along with the corresponding flag

Step 2 - Create the object for accessing the
given web site and the corresponding
regular for the web site.

Step 3 - Check whether the given URL
address with the regular of
the page is not equal to
None then display the message
from URL address else not
found.

Ans

Constitutive expression of *CesA8*

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

in *Arabidopsis* roots

is not required for

cell wall synthesis

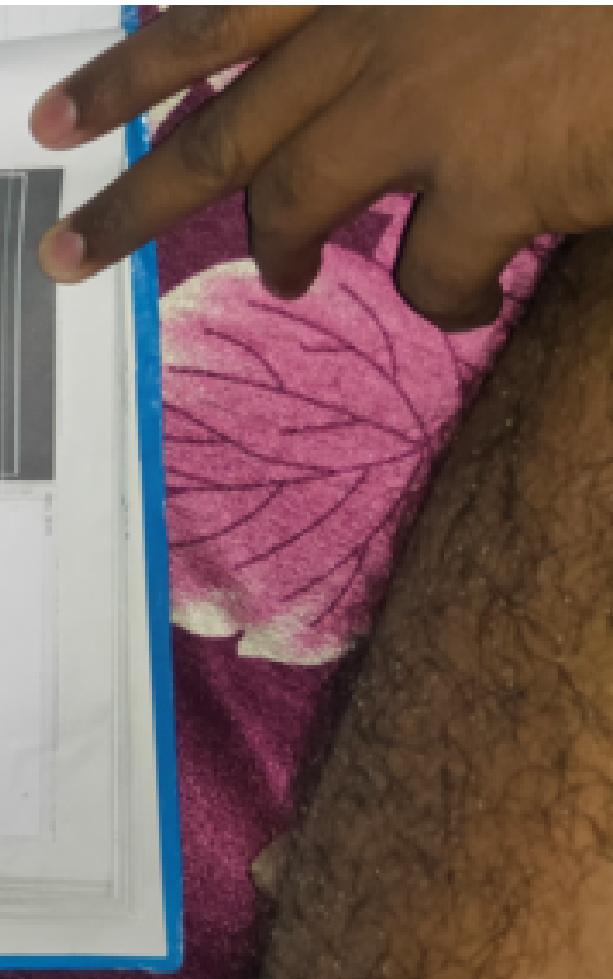
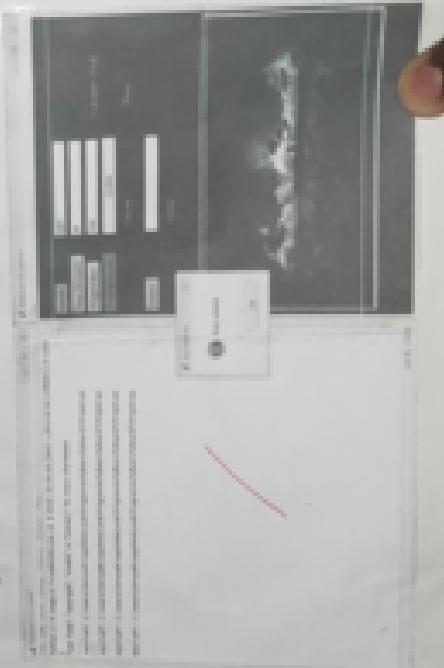
in *Arabidopsis* roots

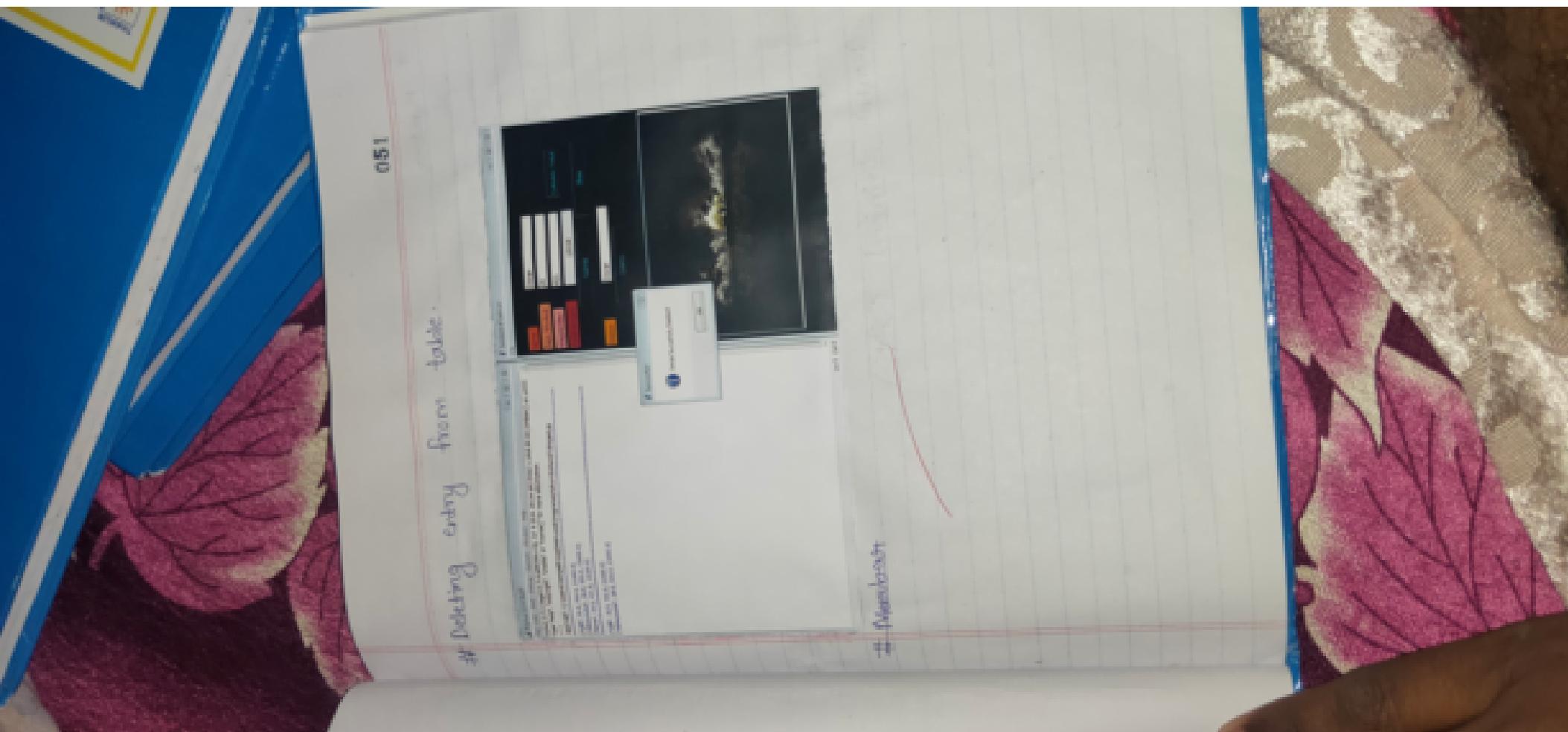
is not required for

cell wall synthesis



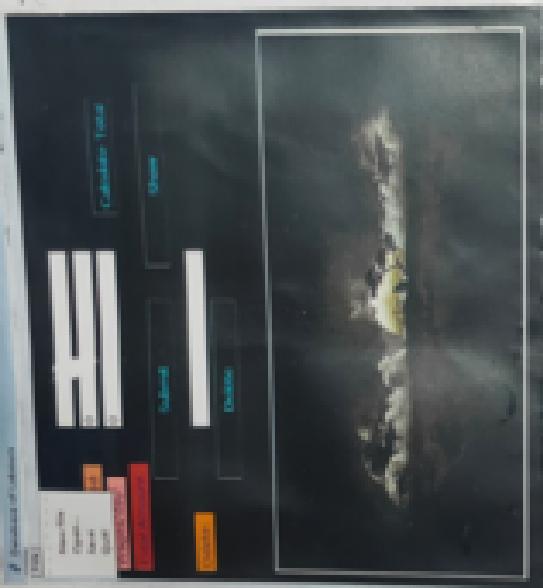
Welding Cracks in Table.







16. *Musica diurna* (Musica diurna)



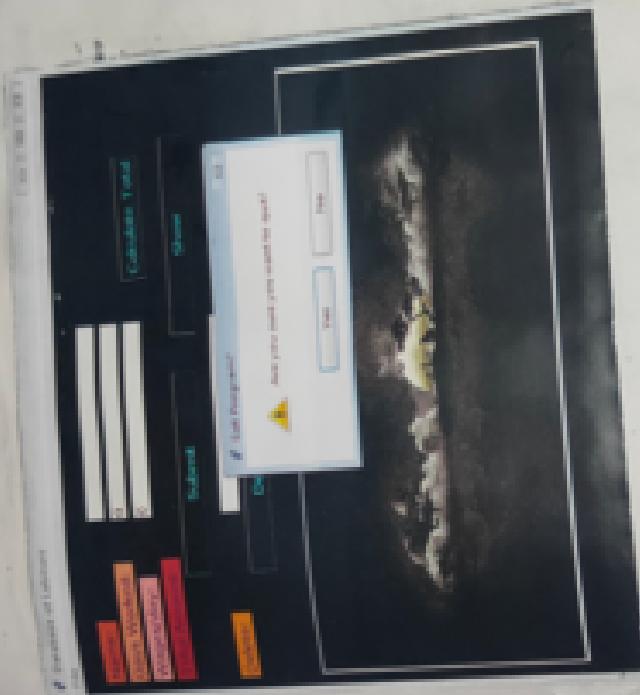
17. *Musica nocturna*.

18.



052
Alypiidae, Contraidae, Geometridae

052



052

