**PRACTICAL 1.**

**AIM: PROGRAMS TO READ AND WRITE FILES**

**1)#Reading from files:**

inputfilename=input("enter the name of input file to read the data from :")

inputfile=open(inputfilename,'r')

print("opening file ",inputfilename ," for readng")

for line in inputfile:

print(line)

inputfile.close()

**OUTPUT:**

enter the name of input file to read the data from :NOTE.txt

opening file NOTE.txt for readng

HELLO, WAP TO READ FROM FILES

>>>

**2) Using with statement:**

with open("myfile.txt","r")as file:

data=file.read();

print(data

**OUTPUT:**

HELLOOOOO

**3)Program to read and write files:**

inputfilename=input("Enter the name of input file to read the grades from :")

outputfilename=input("Enter the name of the output file to record the GPA's to :")

inputfile=open(inputfilename,"r")

outputfile=open(outputfilename,"w")

gpa=0

for line in inputfile:

if(line[0]=="A"):

gpa=4

elif(line[0]=="B"):

gpa=3

elif(line[0]=="C"):

gpa=2

elif(line[0]=="D"):

gpa=1

elif(line[0]=="E"):

gpa=0

else:

gpa=-1

temp=str(gpa)

temp=temp+"\n"

print(line[0],"\t",gpa)

outputfile.write(temp)

inputfile.close()

outputfile.close()

**OUTPUT**

Enter the name of input file to read the grades from :grades.txt

Enter the name of the output file to record the GPA's to :gpa.txt

B 3

O -1

A 4

C 2

D 1

**4)Program to write contents to a file:**

file=input("Enter the file name to write the data to :")

fo=open(file,"w")

data=input("Enter some text/content to write in a file :")

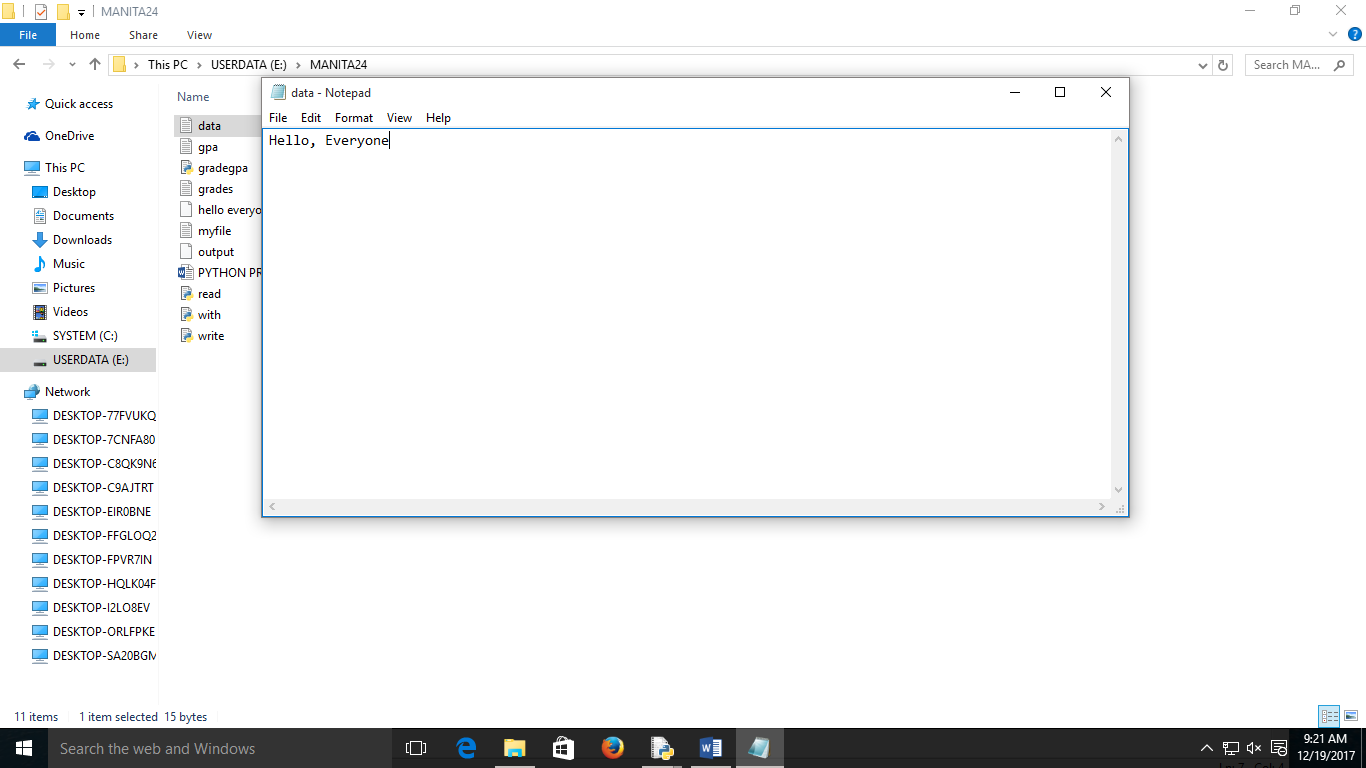
fo.write(data)

fo.close()

**OUTPUT:**

Enter the file name to write the data to :data.txt

Enter some text/content to write in a file :Hello, Everyone



**5)Program to read and write file.**

def myfile():

file=input("Enter the file name to write the data to :")

f1=open(file,"w")

data=("Writer's write, but fingers don't fing,\nNeither apple nor pine are in pineapple ,\nBoxing rings are square,\nA house can burn up as it can burns down,\nOverlook and oversee are opposites,\nAn alarm goes off by going on.")

f1.write(data)

myfile()

filename=input("Enter the filename to read the data from :")

f2=open(filename,"r")

for line in f2:

print(line)

f2.close()

filename=input("Enter the file name to append the data :")

f3=open(filename,"a")

data=input("Enter the data to append in the file ")

f3.write(data)

f3.close()

**OUTPUT:**

Enter the file name to write the data to :append

Enter the filename to read the data from :append

Writer's write, but fingers don't fing,

Neither apple nor pine are in pineapple ,

Boxing rings are square,

A house can burn up as it can burns down,

Overlook and oversee are opposites,

An alarm goes off by going on.

Enter the file name to append the data :append

Enter the data to append in the file good one

**6) Program to display firstline from 10th character onwards.**

with open("data.txt")as file:

lines=file.readlines()

print(lines[0][10:])

**OUTPUT**:

rite, but fingers don't fing,

**7)Program to check the first line of the file**

import fileinput

for var in fileinput.input("data.txt"):

print(fileinput.filelineno(),var)

print(fileinput.isfirstline())

**OUTPUT:**

1 Writer's write, but fingers don't fing,

True

2 Neither apple nor pine are in pineapple ,

False

3 Boxing rings are square,

False

4 A house can burn up as it can burns down,

False

5 Overlook and oversee are opposites,

False

6 An alarm goes off by going on.

False

**8)Program to display last line of file**

data=open("data.txt", 'r')

lastline = ""

for line in data:

lastline = line

print(lastline)

**OUTPUT:**

An alarm goes off by going on.

**9)Write a program to print last line using readlines**

file=open("data.txt","r")

for line in file:

print(file.readlines()[-1])

**OUTPUT**

An alarm goes off by going on.

**PRACTICAL 2**

**AIM: PROGRAMS WITH ITERABLES AND ITERATORS**

>>> from itertools import count

>>> counter=count(start=13)

>>> next(counter)

13

>>> next(counter)

14

----------------------------------------------------------------------------------------------

>>> from itertools import cycle

>>> colors=cycle(['red','white','blue'])

>>> next(colors)

'red'

>>> next(colors)

'white'

>>> next(colors)

'blue'

>>> next(colors)

'red'

--------------------------------------------------------------------------------------------

>>> from itertools import islice

>>> colors=cycle(['red','white','blue'])

>>> limited=islice(colors,0,4)

>>> for x in limited:

print(x)

red

white

blue

red

>>>

PRACTICAL 3

**AIM: PROGRAM TO DEMONSTRATE EXCEPTION HANDLING**

**1)Write a function to compute division and use try/except to catch the exceptions.**

def division(x,y):

try:

return x/y

except ZeroDivisionError:

print(“OOPs......division by zero is not possible”)

ans=division(4,2)

print(ans)

ans1=division(4,0)

print(ans1)

ans3=division(33,11)

print(ans3)

**OUTPUT:**

2.0

OOPs......division by zero is not possible

None

3.0

**2)Write a function to enter the age of the user and use try/except to catch the ValueError exception if the age is less than zero.**

def ageChk():

try:

age=int(input(“enter your age))

if(age&lt;0):

raise ValueError

except ValueError:

print(“OOPs......age less then zero not possible”)

finally:

print(“ITS DONE”)

ageChk()

**OUTPUT**

enter your age5

ITS DONE

enter your age-1

OOPs......age less then zero not possible

ITS DONE

def division(x,y):

try:

return x/y

except ZeroDivisionError:

print("division by zero")

division(5,2)

division(5,0)

**OUTPUT:**

division by zero

**Program for demonstration of raise statement**

def div(x,y):

try:

if y==0: raise ZeroDivisionError

return x/y

except ZeroDivisionError:print("zero")

div(3,4)

div(3,0)

**OUTPUT**

zero

----------------------------------------------

fridge\_content={"Egg":8,"Mushroom":20,"Pepper":3,"Cheese":2,"Tomato":4,"Milk":13}

try:

if fridge\_content["orange juice"]>3:

print("Sure,lets have some juice!")

except KeyError:

print("Aww,there is no juice.Let's go shopping!")

**OUTPUT:**

Aww,there is no juice.Let's go shopping!

----------------------------------------------

fridge\_content={"Egg":8,"Mushroom":20,"Pepper":3,"Cheese":2,"Tomato":4,"Milk":13}

try:

if fridge\_content["orange juice"]>3:

print("Sure, let’s have some juice")

except (KeyError)as error:

print("Woah! There is no %s"% error)

except (KeyError, TypeError)as error:

print("Woah! There is no %s"% error)

**OUTPUT:**

Woah! There is no 'orange juice'

----------------------------------------------------------------------------------------

fridge\_content={"Egg":8,"Mushroom":20,"Pepper":3,"Cheese":2,"Tomato":4,"Milk":13}

try:

if fridge\_content["orange juice"] > 3:

print("Sure, let’s have some juice")

except (KeyError) as error:

print("Woah! There is no %s"% error)

except (TypeError):

pass

**OUTPUT:**

Woah! There is no 'orange juice'

**PRACTICAL 4**

**AIM: PROGRAM TO DEMONSTRATE THE USE OF REGULAR EXPRESSION**

**6.1:**

import re

print("Original string: Global Warming")

print("a:To display last 4 characters")

a='Global Warming'

b=re.search('....$',a)

print(b.group(0))

print("\nb:To display substring starting from index 4 and ending at index 8")

b= re.search('Glob(.+?)rming',a)

print(b.group(1))

print("\nc:To check whether string has alphanumeric characters or not.")

r=re.match('[0-9a-zA-Z]+',a)

if r==None:

print("String doesen't contain alphanumeric characters")

else:

print("String contains alphanumeric characters")

print("\nd:To replace all occurances of a with \*")

b=re.sub(r'a','\*',a)

print(b)

print("\ne:To display the starting index for the substring 'Wa'")

print(re.search("Wa",a).start())

print("\nf:To check if the string is in title case")

r=re.match("[A-Z]",a)

if r==None:

print("String is not in title case")

else:

print("String is in title case")

a='Savarkar road 400060'

print("String:",a)

print("\nfind the digits")

print(re.findall(r'\d',a))

print("\nfind the non-digits")

print(re.findall(r'\D',a))

print("\nfind the non-space")

print(re.findall(r'\S',a))

**Output:**

Original string: Global Warming

a:To display last 4 characters

ming

b:To display substring starting from index 4 and ending at index 8

al Wa

c:To check whether string has alphanumeric characters or not.

String contains alphanumeric characters

d:To replace all occurances of a with \*

Glob\*l W\*rming

e:To display the starting index for the substring 'Wa'

7

f:To check if the string is in title case

String is in title case

String: Savarkar road 400060

find the digits

['4', '0', '0', '0', '6', '0']

find the non-digits

['S', 'a', 'v', 'a', 'r', 'k', 'a', 'r', ' ', 'r', 'o', 'a', 'd', ' ']

find the non-space

['S', 'a', 'v', 'a', 'r', 'k', 'a', 'r', 'r', 'o', 'a', 'd', '4', '0', '0', '0', '6', '0']

>>>

**6.2:**

import re

line = "Cats are smarter than dogs"

matchObj = re.match( r'(.\*) are (.\*?) .\*', line, re.M|re.I)

if matchObj:

print("matchObj.group():", matchObj.group())

print("matchObj.group(1):", matchObj.group(1))

print("matchObj.group(2):", matchObj.group(2))

else:

print("No match!!")

searchObj = re.search( r'(.\*) are (.\*?) .\*', line, re.M|re.I)

if searchObj:

print("searchObj.group():", searchObj.group())

print("searchObj.group(1):", searchObj.group(1))

print("searchObj.group(2):", searchObj.group(2))

else:

print("Nothing found!!")

print("Matching VS Searching:")

matchObj = re.match( r'dogs', line, re.M|re.I)

if matchObj:

print ("match --> matchObj.group() : ", matchObj.group())

else:

print( "No match!!")

searchObj = re.search( r'dogs', line, re.M|re.I)

if searchObj:

print ("search --> searchObj.group() : ", searchObj.group())

else:

print ("Nothing found!!")

print("re.sub():")

phone = "2004-959-559 # This is Phone Number"

# Delete Python-style comments

num = re.sub(r'#.\*$', "hi", phone)

print ("Phone Num :", num)

# Remove anything other than digits

num = re.sub(r'\D', "", phone)

print ("Phone Num :", num)

**Output:**

>>>

matchObj.group(): Cats are smarter than dogs

matchObj.group(1): Cats

matchObj.group(2): smarter

searchObj.group(): Cats are smarter than dogs

searchObj.group(1): Cats

searchObj.group(2): smarter

Matching VS Searching:

No match!!

search --> searchObj.group() : dogs

re.sub():

Phone Num : 2004-959-559 hi

Phone Num : 2004959559

**In this example a string is given and we are matching the full string first string and the second string**

>>> import re

>>> line="humans are itelligent than animals"

>>> so=re.match(r"(.\*) are (.\*?) .\*",line,re.M|re.I)

>>> if so:

print("so.group():",so.group())

print("so.group():",so.group(1))

print("so.group():",so.group(2))

else:

print("no match found")

output:

so.group(): humans are itelligent than animals

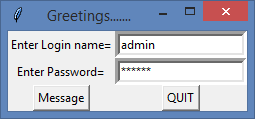
so.group(): humans

so.group(): intelligent

**PRACTICAL 5**

**AIM: PROGRAMS TO SHOW DRAW SHAPES AND GUI CONTROLS**

* 1. Write a python program to Design a Login Form containing User Name and password. When user name and password matches with predefined values, a welcome message should be displayed otherwise error message should be displayed. If wrong passwords is entered, the application should end.[Use If]



**Program:**

from tkinter import\*

from tkinter import messagebox

top=Tk()

def cancel():

top.destroy()

def save():

a=x.get()

b=y.get()

if(a=="fybsccs1234@gmail.com" and b=="fycs1234"):

messagebox.showinfo("login","u have logged in")

else:

messagebox.showinfo("attention","something went wrong")

top.title("login portal")

x=StringVar()

y=StringVar()

l1=Label(top,text="enter your email addr").grid(row=0,column=0)

l2=Label(top,text="enter your passwd").grid(row=1,column=0)

e1=Entry(top,textvariable=x).grid(row=0,column=1)

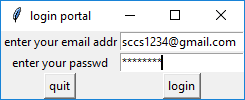
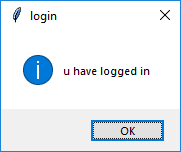
e2=Entry(top,textvariable=y,show='\*').grid(row=1,column=1)

b1=Button(top,text="quit",command=cancel).grid(row=2,column=0)

b2=Button(top,text="login",command=save).grid(row=2,column=1)

top.mainloop()

**output:**

* 1. Write a python program that accepts Principle Amount, No. of Years & Rate of Interest from 3 text fields, when you click “Calculate Interest” button, the data is sent to a function that returns the simple interest. When you click on “Final Amount” button, the final amount by adding principle amount and interest should be displayed. (H.W)

**PROGRAM**

from tkinter import \*

from tkinter import messagebox

root=Tk()

root.title('simple interest')

def ci():

a=x.get()

b=y.get()

c=z.get()

v=(a\*b\*c)/100

messagebox.showinfo('simple interest is',v)

def final():

p=x.get()

q=y.get()

w=p+q

messagebox.showinfo('final amount is',w)

x=IntVar()

y=IntVar()

z=IntVar()

l1=Label(root,text='Principle amount').grid(row=0,column=0)

e1=Entry(root,textvariable=x).grid(row=0,column=1)

l2=Label(root,text='Rate of interest').grid(row=1,column=0)

e2=Entry(root,textvariable=y).grid(row=1,column=1)

l3=Label(root,text='No. of years').grid(row=2,column=0)

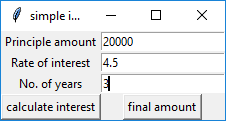
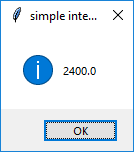
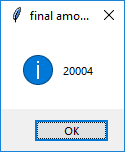
e3=Entry(root,textvariable=z).grid(row=2,column=1)

b1=Button(root,text='calculate interest',command=ci).grid(row=3,column=0)

b2=Button(root,text='final amount',command=final).grid(row=3,column=1)

root.mainloop()

**output:-**

* 1. Write a python program to Create a reservation form for Mumbai-Pune journey, containing 3 text-fields to enter names, 3 text-fields to enter age and a text field to show final bill. It should also contain a check box showing the type of journey(AC or NonAC). Charges of AC/NonAC mode of journey is fixed. But only for children(age <5) and senior citizens(age>60), the rates are half. 4% service charges are applied on final amount. As per the passengers entered by user, display the final bill. (H.W)

**PROGRAM**

from tkinter import \*

def ac():

age1=x.get()

age2=y.get()

age3=z.get()

amount=500

bill=0

if(age1>60 or age1<5):

bill=bill+(amount/2)

else:

bill=bill+amount

if(age2>60 or age2<5):

bill=bill+(amount/2)

else:

bill=bill+amount

if (age3>60 or age3<5):

bill=bill+(amount/2)

else:

bill=bill+amount

ans.set(bill)

def nonac():

age1=x.get()

age2=y.get()

age3=z.get()

amount=300

bill=0

if(age1>60 or age1<5):

bill=bill+(amount/2)

else:

bill=bill+amount

if(age2>60 or age2<5):

bill=bill+(amount/2)

else:

bill=bill+amount

if (age3>60 or age3<5):

bill=bill+(amount/2)

else:

bill=bill+amount

ans.set(bill)

root=Tk()

root.title("Mumbai Pune ticket Reservation")

l1=Label(root,text="enter pessanger name 1:").grid(row=0,column=0)

l2=Label(root,text="enter pessanger name 2:").grid(row=1,column=0)

l3=Label(root,text="enter pessanger name 3:").grid(row=2,column=0)

l4=Label(root,text="age 1:").grid(row=0,column=2)

l5=Label(root,text="age 2:").grid(row=1,column=2)

l6=Label(root,text="age 3:").grid(row=2,column=2)

l7=Label(root,text="final bill amount:").grid(row=3,column=0)

x=IntVar()

y=IntVar()

z=IntVar()

ans=IntVar()

abc=IntVar()

b1=Radiobutton(root,text="ac",variable=abc,value=1,command=ac).grid(row=4,column=0)

b2=Radiobutton(root,text="Non-ac",variable=abc,value=2,command=nonac).grid(row=5,column=0)

c1=Entry(root).grid(row=0,column=1)

c2=Entry(root).grid(row=1,column=1)

c3=Entry(root).grid(row=2,column=1)

c4=Entry(root,textvariable=x).grid(row=0,column=3)

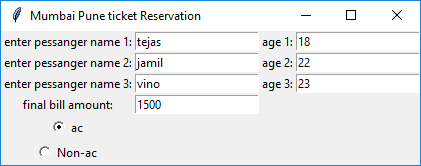
c5=Entry(root,textvariable=y).grid(row=1,column=3)

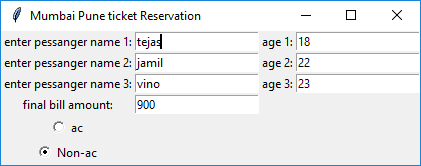
c6=Entry(root,textvariable=z).grid(row=2,column=3)

c7=Entry(root,textvariable=ans).grid(row=3,column=1)

root.mainloop()

**output**





* 1. Write a python program to Create a form containing a list box with names of Indian cricket players. The program should perform following operations –

insertion of new elements in to the list,

deletion of selected element from the list,

show the currently selected element,

show total count

**PROGRAM:**

from tkinter import \*

from tkinter import messagebox

root = Tk()

def addnew():

data=x.get()

s=l1.size()

l1.insert(s,data)

messagebox.showinfo("insertion","insertion done")

def delete():

l1.delete(l1.curselection())

messagebox.showinfo("deletion","deletion done")

def display():

messagebox.showinfo("you have selected",l1.get(l1.curselection()))

def totalcount():

s=l1.size()

messagebox.showinfo("total count",s)

root.title("INDIAN CRICKET PLAYERS")

x=StringVar()

l1 = Listbox(root)

l1.grid(row=0,column=0)

l1.insert(0,"DON BRADMAN")

l1.insert(1,"GARRY SOBERS")

l1.insert(2,"VIVIAN RICHARDS")

l1.insert(3,"DESNEL HEYNES")

l1.insert(4,"SACHIN TENDULKAR")

l1.insert(5,"RICHARD HADLEY")

l1.insert(6,"SUNIL GAVASKAR")

l1.insert(7,"STEVE WARNE")

l1.insert(8,"RAHUL DRAVID")

l1.insert(9,"ANIL KUMBLE")

l1.insert(10,"MALCOM MARSHALL")

e1=Entry(root,textvariable=x).grid(row=1,column=0)

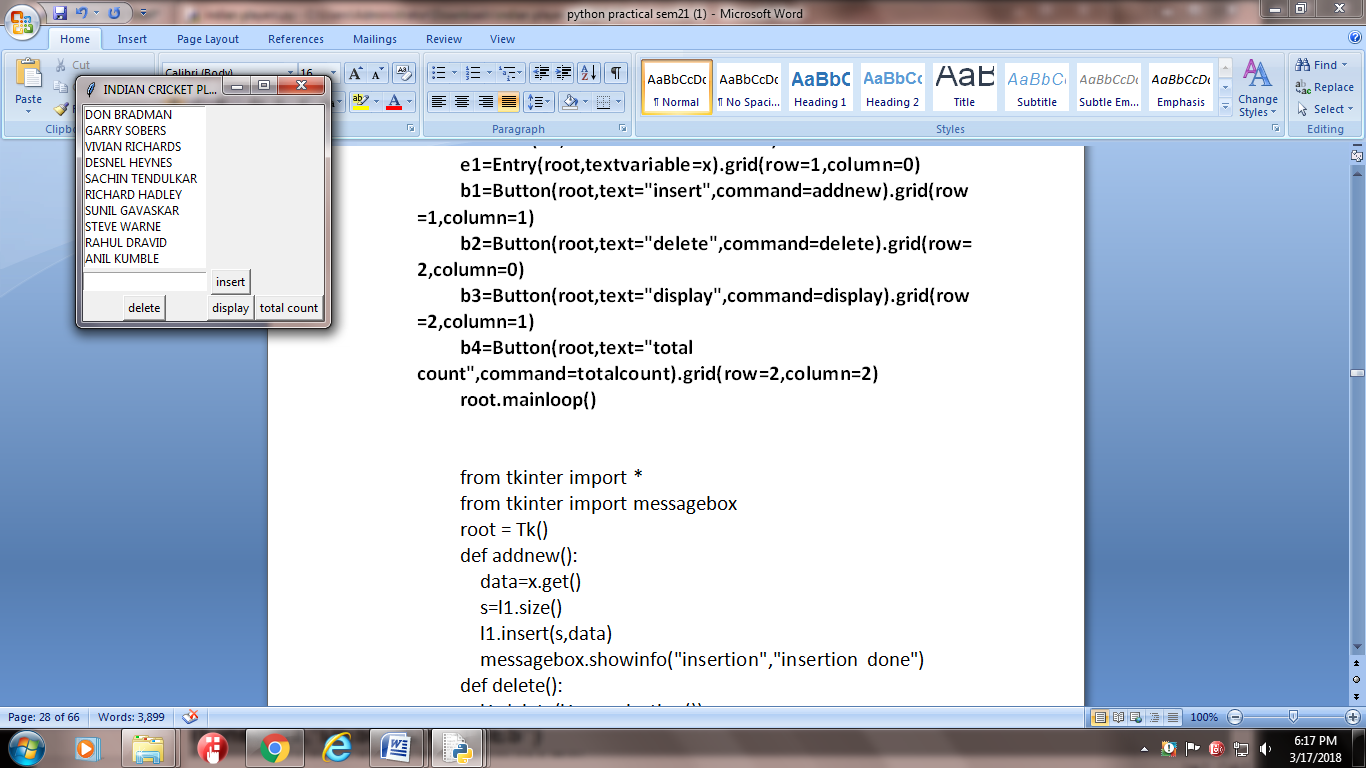
b1=Button(root,text="insert",command=addnew).grid(row=1,column=1)

b2=Button(root,text="delete",command=delete).grid(row=2,column=0)

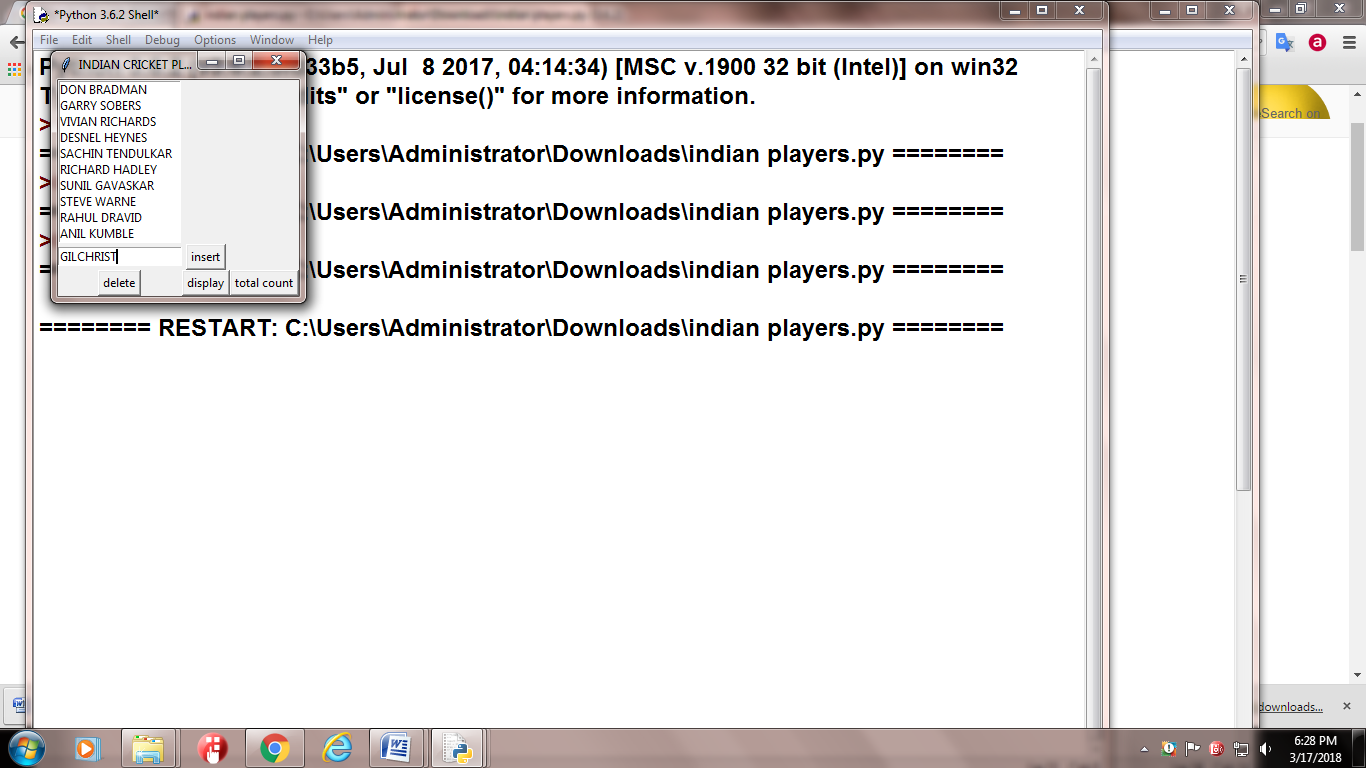
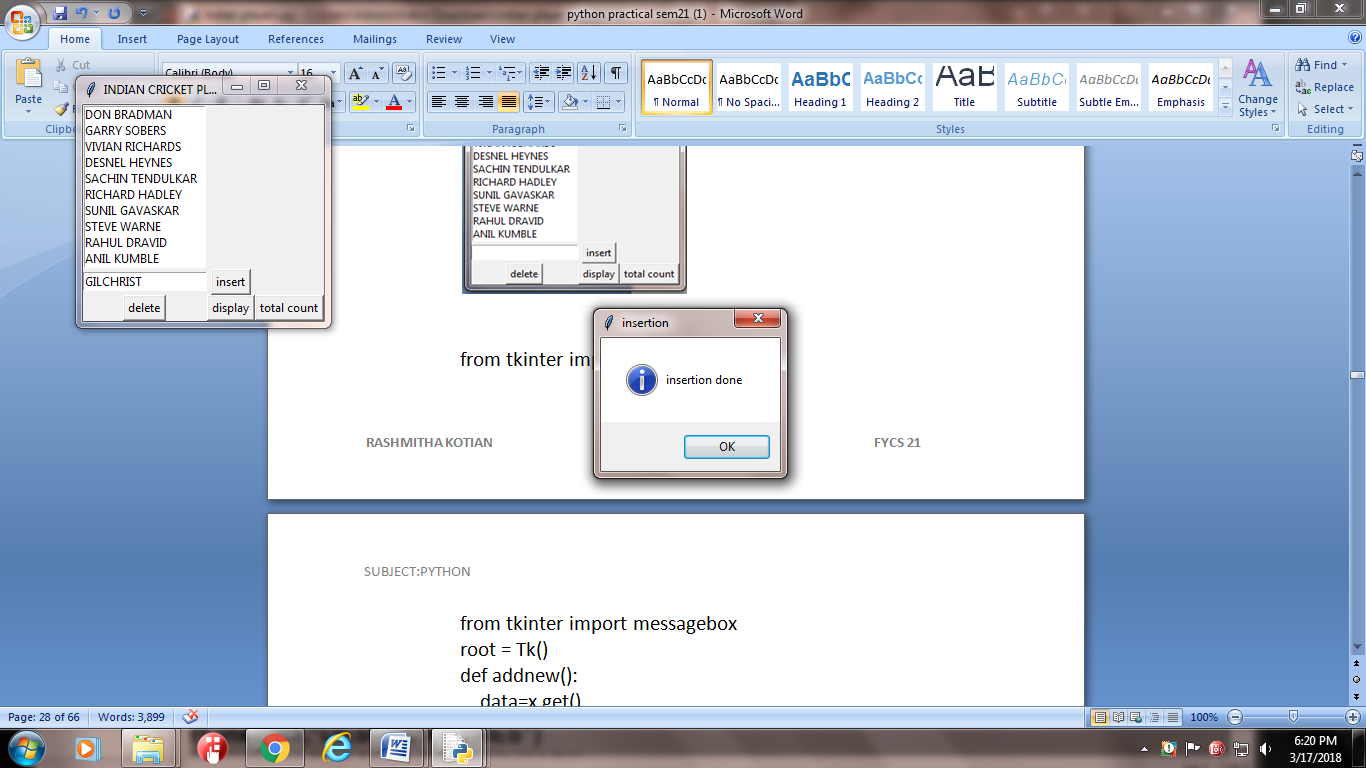
b3=Button(root,text="display",command=display).grid(row=2,column=1)

b4=Button(root,text="total count",command=totalcount).grid(row=2,column=2)

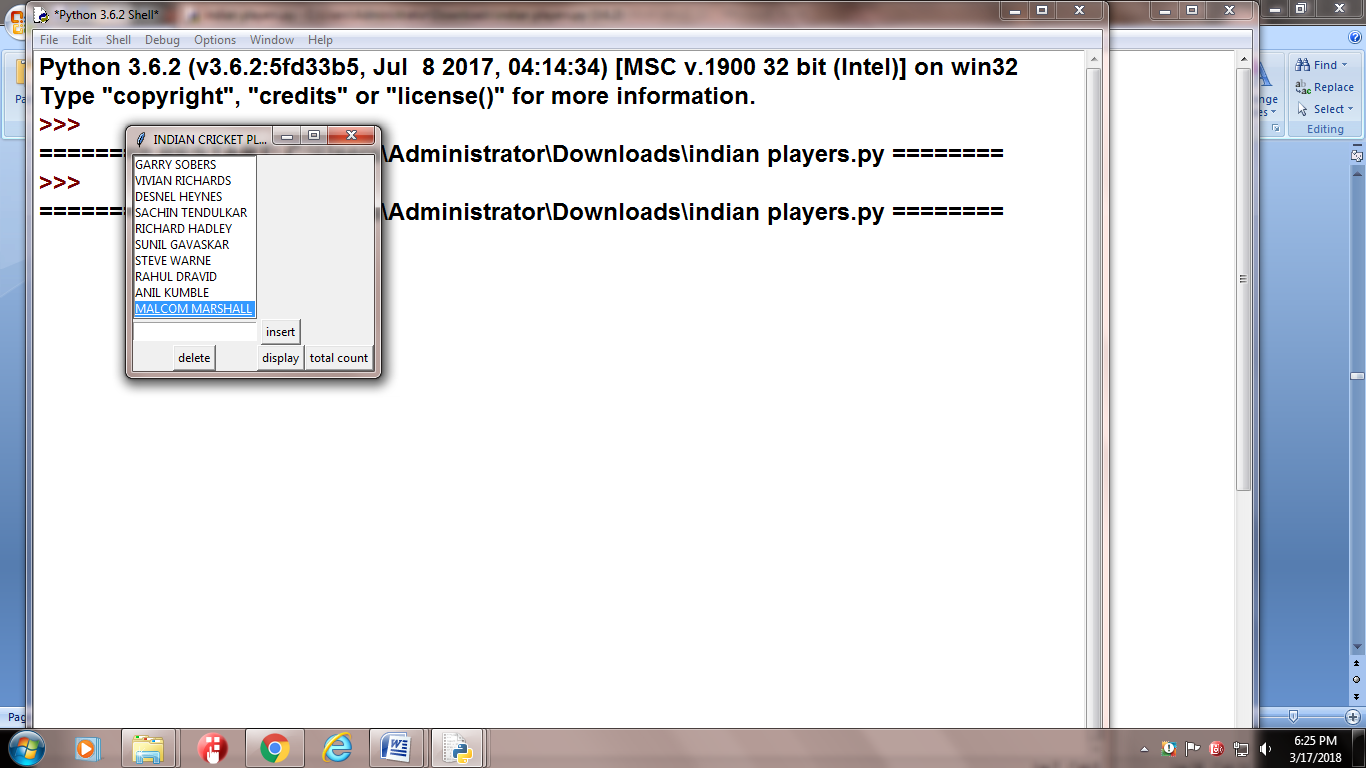
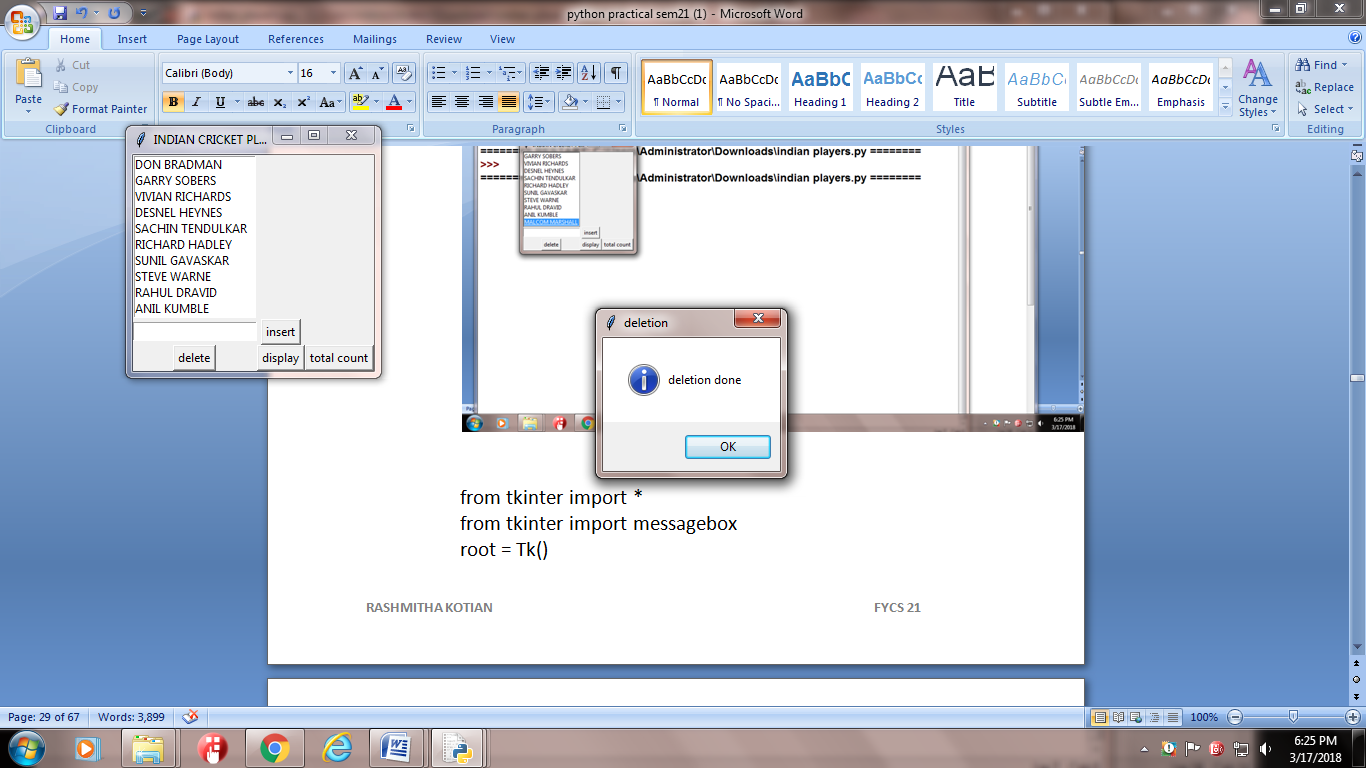
root.mainloop()



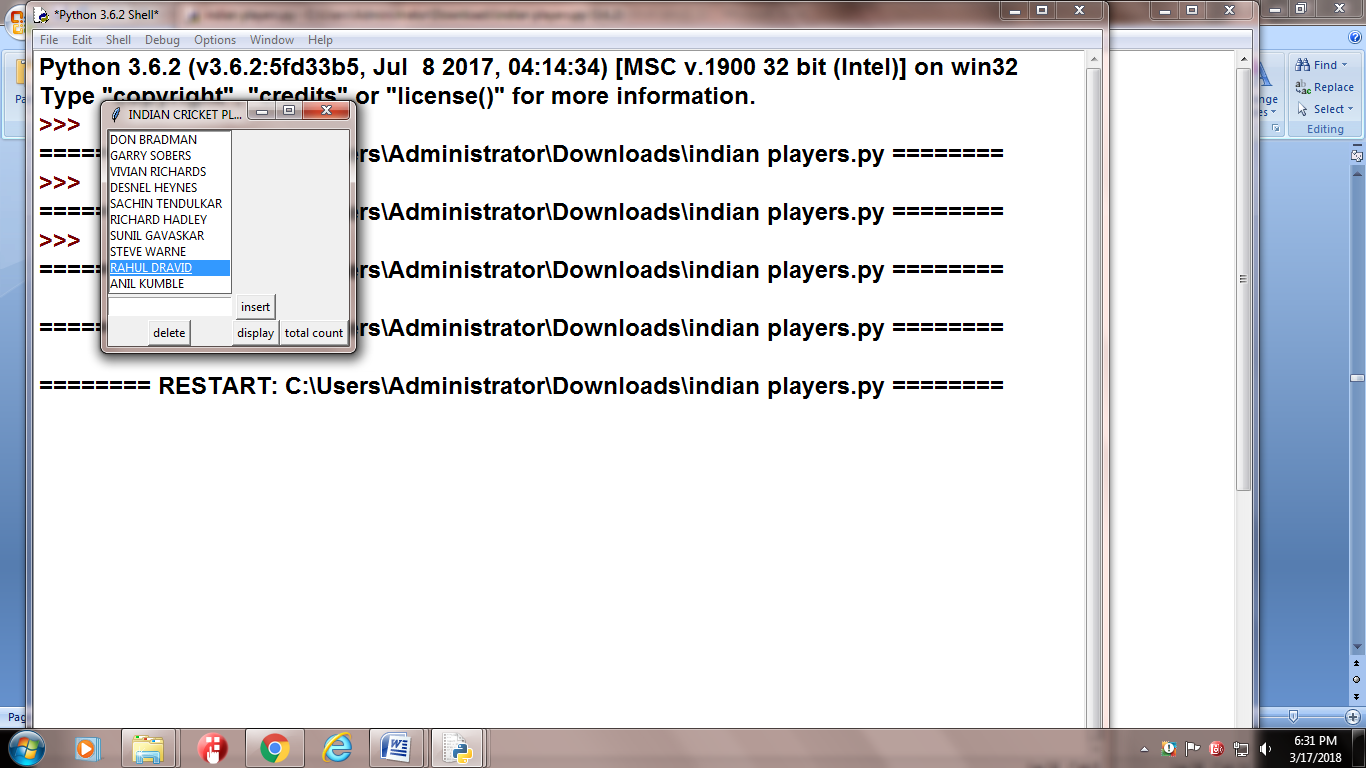
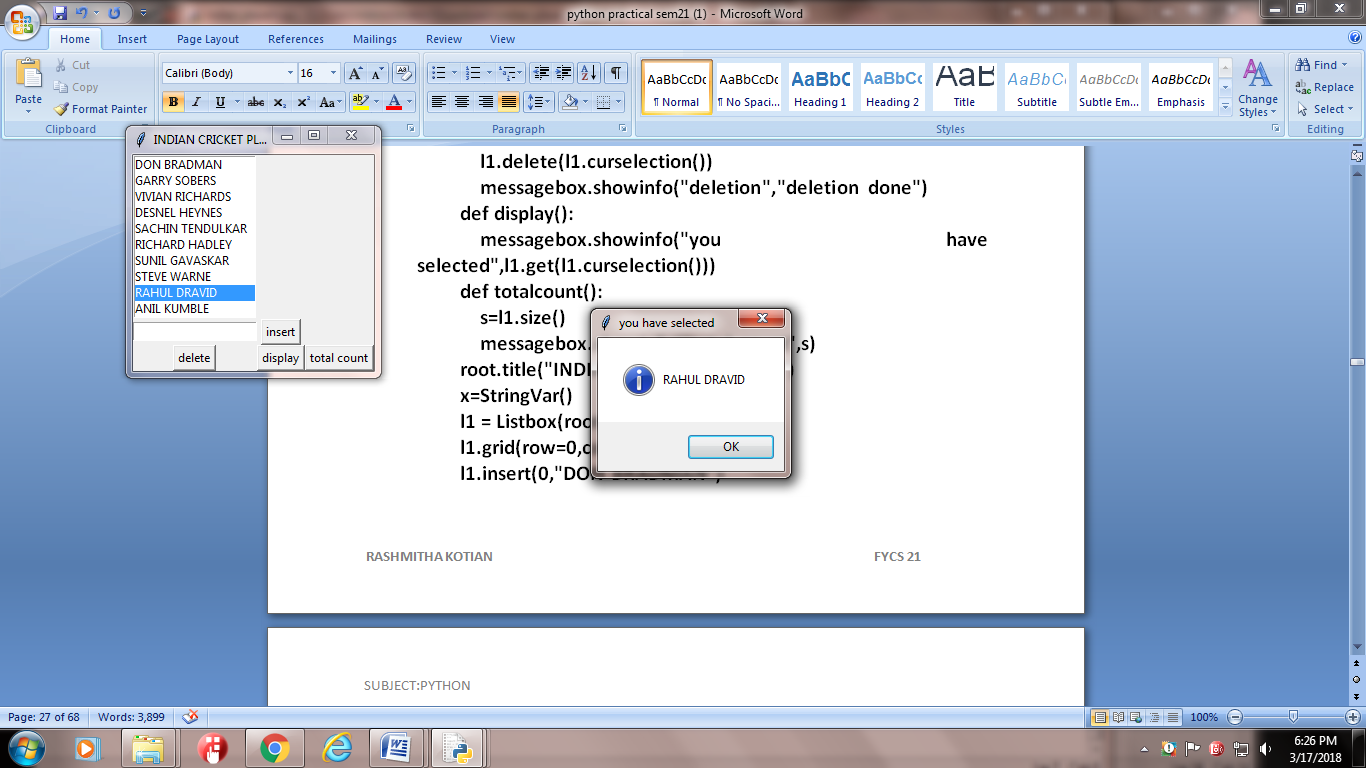
**1)INSERTION:**

 ****

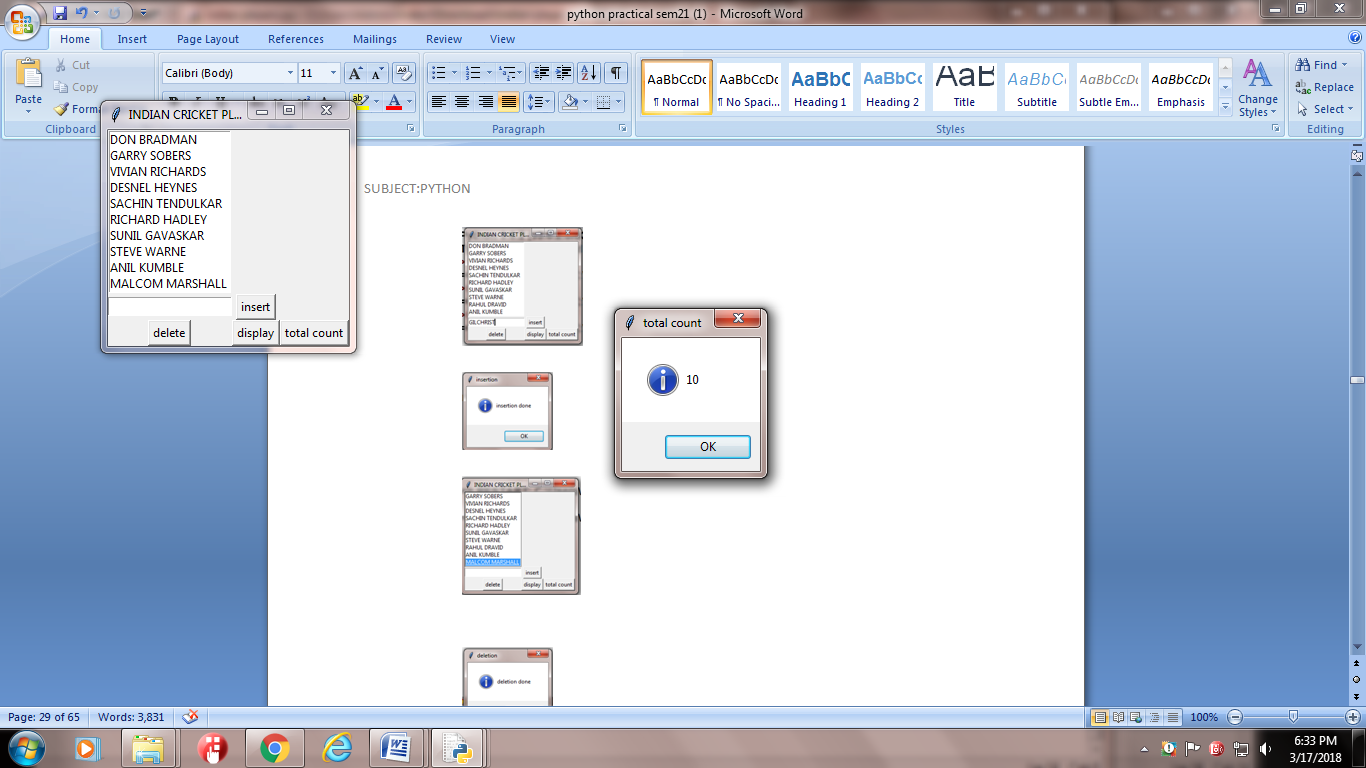
**2)DELETION:**

 ****

**3)DISPLAY:**

**4)COUNT:**



* 1. Write a program to accept information of customer for an investment policy such as Name, Date of Birth and Address. The amount to be invested should be selected from a combo box. A customer can invest from min. age of 21 to 60. According to Date of birth, determine current age of customer, total number of years remaining and accordingly find & display the premium that customer will have to pay. (H.W)

**PROGRAM**

from tkinter import\*

from tkinter import messagebox

from tkinter.ttk import \*

from datetime import \*

def cal\_age():

dob=y.get()

dt1=datetime.strptime(dob,'%d/%m/%Y')

dt=datetime.now()

dt2=datetime.strftime(dt,'%d/%m/%Y')

ans=dt.year-dt1.year

if (ans<21 or ans>60):

messagebox.showinfo("attention","age is not satisfied")

else:

messagebox.showinfo("your age is:",ans)

def cal\_pre():

yrs=c.get()

amt=int(cb.get())

prem=(amt/yrs)

messagebox.showinfo("amt is",prem)

top=Tk()

top.title("investment policy")

l1=Label(top,text="customer name").grid(row=0,column=0)

l2=Label(top,text="date of birth(dd/mm/yyyy)").grid(row=1,column=0)

l3=Label(top,text="address").grid(row=2,column=0)

l4=Text(top,width=12,height=5).grid(row=2,column=1)

l5=Label(top,text="duration of investment(years)").grid(row=3,column=0)

l6=Label(top,text="investment amt").grid(row=4,column=0)

x=StringVar()

y=StringVar()

c=IntVar()

d=IntVar()

e1=Entry(top,textvariable=x).grid(row=0,column=1)

e2=Entry(top,textvariable=y).grid(row=1,column=1)

e3=Entry(top,textvariable=c).grid(row=3,column=1)

cb=Combobox(top,width=10,height=10)

cb['values']=(200000,300000,1000000,5000000)

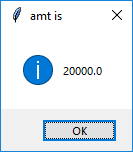
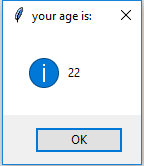
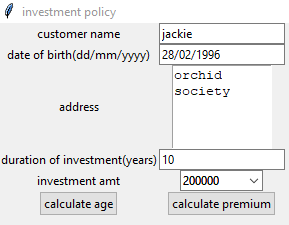
cb.grid(row=4,column=1)

b1=Button(top,text="calculate age",command=cal\_age).grid(row=5,column=0)

b2=Button(top,text="calculate premium",command=cal\_pre).grid(row=5,column=1)

top.mainloop()

**OUTPUT:**



6)A software company wants to conduct placements for graduates in Mumbai or Pune. Design a PYTHON interface to input student details such as name using textbox, course name such as BSc Comp Sci/IT/Phy/Maths using radio button, Marks in SSC,HSC, FY & SY along with Year of Passing using text boxes & Location such as Mumbai/Pune using combo box. The student is shortlisted only if he/she has min 60% in SSC & HSC, FY & SY average should be min 55% & there should not be any gap in education. When student clicks on Register button, informs the student whether he/she qualifies and place of interview in Mumbai or Pune as given by the student. (H.W)

**PROGRAM**

from tkinter import \*

from tkinter import messagebox

from tkinter.ttk import \*

def check():

a1=y.get()

b1=z.get()

c1=a.get()

d1=b.get()

avg=(c1+d1)/2

y1=c.get()

y4=f.get()

yrs=(y4-y1)

city=cb.get()

if (a1>=60 and b1>=60 and avg>=55):

if(yrs==4):

if(city=="Pune"):

messagebox.showinfo("Qualified","you are eligible for placement\n Your venue for placement is sidhharth college pune")

else:

messagebox.showinfo("Qualified","you are eligible for placement\n Your venue for placement is bhavans college mumbai")

else:

messagebox.showinfo("Not Qualified","Not eligible because you have a gap in education")

else:

messagebox.showinfo("Not Qualified","Not eligible in percentage")

root=Tk()

root.title("Placement Details")

x=StringVar()

y=IntVar()

z=IntVar()

a=IntVar()

b=IntVar()

c=IntVar()

d=IntVar()

e=IntVar()

f=IntVar()

r=IntVar()

l1=Label(root,text="Enter your name") .grid(row=0,column=0)

e1=Entry(root,textvariable=x).grid(row=0,column=1)

l2=Label(root,text="Course name").grid(row=1,column=0)

r1= Radiobutton(root, text="BSc comp sci",variable=r, value=1).grid(row=1,column=1)

r2= Radiobutton(root, text="Bsc IT", variable=r,value=2).grid(row=1,column=2)

r3= Radiobutton(root, text="BSc phy", variable=r,value=3).grid(row=1,column=3)

r4= Radiobutton(root, text="Bsc maths", variable=r,value=4).grid(row=1,column=4)

l3=Label(root,text="Marks details").grid(row=2,column=1)

l4=Label(root,text="Year of passing").grid(row=2,column=2)

l5=Label(root,text="marks in ssc").grid(row=3,column=0)

e2=Entry(root,textvariable=y).grid(row=3,column=1)

l6=Label(root,text="marks in hsc").grid(row=4,column=0)

e3=Entry(root,textvariable=z).grid(row=4,column=1)

l7=Label(root,text="marks in fy").grid(row=5,column=0)

e4=Entry(root,textvariable=a).grid(row=5,column=1)

l8=Label(root,text="marks in sy").grid(row=6,column=0)

e5=Entry(root,textvariable=b).grid(row=6,column=1)

e6=Entry(root,textvariable=c).grid(row=3,column=2)

e7=Entry(root,textvariable=d).grid(row=4,column=2)

e8=Entry(root,textvariable=e).grid(row=5,column=2)

e9=Entry(root,textvariable=f).grid(row=6,column=2)

l9=Label(root,text="Location").grid(row=7,column=0)

cb=Combobox(root,width=5,height=4)

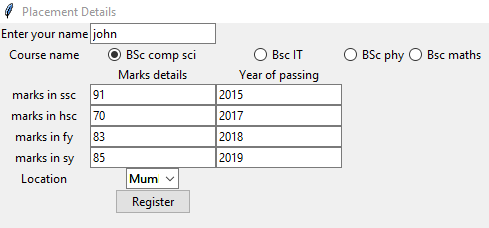
cb["values"]=("Mumbai","Pune")

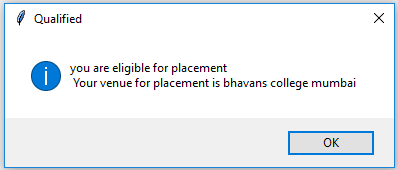
cb.grid(row=7,column=1)

b1=Button(root,text="Register",command=check).grid(row=8,column=1)

root.mainloop()

**output**





1. A bank wants to keep an enquiry page for inquiring about home loan. The rates of these loans are 7.8, 7.9 AND 9.8 for various span of years (5/10/15) respectively. The user types the loan amount & selects the number of years for repayment of loan (5/10/15) from combo box. In return, the user gets information of the rate of loan, monthly installment for the specified number of years. (H.W)

**Home loan**

**Input:**

from tkinter import \*

from tkinter.ttk import \*

from tkinter import messagebox

def inst():

amt=a.get()

y=c.get()

f=0

rate=0

if(y==5):

f=amt\*0.78

rate=7.8

elif(y==10):

f=amt\*0.79

rate=7.9

elif(y==15):

f=amt\*0.98

rate=9.8

r=amt+f

if(y==5):

r=r/(5\*12)

elif(y==10):

r=r/(10\*12)

elif(y==15):

r=r/(15\*12)

messagebox.showinfo("Rate of Loan is",rate)

messagebox.showinfo("Monthly Installment is",r)

root=Tk()

c=IntVar()

a=IntVar()

root.title("Home loan enquiry")

l1=Label(root,text="Loan Amount").grid(row=0,column=0)

e1=Entry(root,textvariable=a).grid(row=0,column=1)

l2=Label(root,text="No. of year for repayment").grid(row=1,column=0)

cb=Combobox(root,textvariable=c)

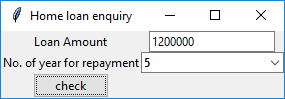
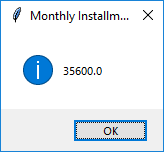
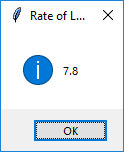
cb["values"]=(5,10,15)

cb.grid(row=1,column=1)

b1=Button(root,text="check",command=inst).grid(row=2,column=0)

root.mainloop()

**OUTPUT:**

8)Design ticket enquiry form for a theater. Select location (Mumbai/Pune) using radio button. When user clicks a location, it fills a combo box with names of theaters in that city. When user selects the name of the theatre, the list of films currently shown and their show timings should be displayed.

**PROGRAM**

from tkinter import \*

from tkinter import messagebox

from tkinter.ttk import \*

root=Tk()

def MUMBAI():

cb["values"]=("IMAX","GLOBAL FAN","CITY MALL","PHOENIX")

def PUNE():

cb["values"]=("PVR","SUNCITY","RCT MALL")

def showdetail():

data=i.get()

if(data=="IMAX"):

messagebox.showinfo("detail","movie name=movie1\ntiming\nscreen1:3pm-6pm")

elif(data=="GLOBAL FAN"):

messagebox.showinfo("detail","movie name=movie2\ntiming\nscreen1:12pm-3pm\nscreen2: 3pm-6pm")

elif(data=="CITY MALL"):

messagebox.showinfo("detail","movie name=movie3\ntiming\nscreen1:3pm-6pm")

elif(data=="PHOENIX"):

messagebox.showinfo("detail","movie name=movie4\ntiming\nscreen1:12pm-3pm\nscreen2: 3pm-6pm")

elif(data=="PVR"):

messagebox.showinfo("detail","movie name=movie5\ntiming\nscreen1:3pm-6pm")

elif(data=="SUNCITY"):

messagebox.showinfo("detail","movie name=movie6\ntiming\nscreen1:12pm-3pm\nscreen2: 3pm-6pm")

elif(data=="RCT MALL"):

messagebox.showinfo("detail","movie name=movie7\ntiming\nscreen1:3pm-6pm")

i=StringVar()

r=StringVar()

root.title("TICKET ENQUIRY FOR THEATRE")

l1=Label(root,text="SELECT LOCATION").grid(row=0,column=0)

l2=Label(root,text="LIST OF THEATERS").grid(row=2,column=0)

r1= Radiobutton(root, text="MUMBAI",variable=r, value=1,command=MUMBAI).grid(row=1,column=0)

r2= Radiobutton(root, text="PUNE", variable=r,value=2,command=PUNE).grid(row=1,column=1)

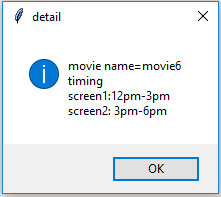
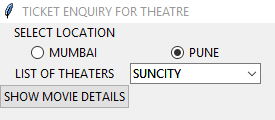
cb=Combobox(root,width=18,height=20,textvariable=i)

cb.grid(row=2,column=1)

b1=Button(root,text="SHOW MOVIE DETAILS",command=showdetail).grid(row=7,column=0)

root.mainloop()

**OUTPUT**



**Write a gui program to perform arithmetic operations on two numbers**

**PROGRAM**

import math

from tkinter import \*

from tkinter import messagebox

def add():

a=x.get()

b=y.get()

messagebox.showinfo("addition",a+b)

def sub():

a=x.get()

b=y.get()

messagebox.showinfo("subtraction",a-b)

def multi():

a=x.get()

b=y.get()

messagebox.showinfo("multiplication",a\*b)

def div():

a=x.get()

b=y.get()

messagebox.showinfo("DIVISION",a/b)

def power():

a=x.get()

b=y.get()

messagebox.showinfo("POWER",a\*\*b)

root=Tk()

root.title("ARITHMETIC OPERATIONS")

x=IntVar()

y=IntVar()

u=Label(root,text="ENTER THE OPERATION")

a=Entry(root,textvariable=x)

b=Entry(root,textvariable=y)

b1=Button(root,text="add",command=add)

b2=Button(root,text="sub",command=sub)

b3=Button(root,text="multi",command=multi)

b4=Button(root,text="div",command=div)

b5=Button(root,text="pow",command=power)

u.pack(side=LEFT)

a.pack(side=RIGHT)

b.pack(side=RIGHT)

b1.pack(side=BOTTOM)

b2.pack(side=BOTTOM)

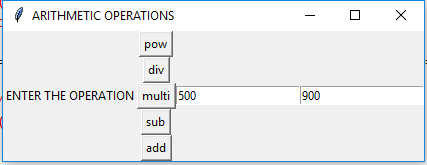
b3.pack(side=BOTTOM)

b4.pack(side=BOTTOM)

b5.pack(side=BOTTOM)

root.mainloop()

**output:**



**Write a gui program to check whether the candidate is eligible for voting or not**

**Program**

from tkinter import \*

from tkinter import messagebox

def checkage():

age=x.get()

if age>=18:

messagebox.showinfo("result","you are eligi ble")

else:

messagebox.showinfo("result","you are not eligible")

root=Tk()

root.title("checking voting eligible")

x=IntVar()

u=Label(root,text="enter your age:")

a=Entry(root,textvariable=x)

b1=Button(root,text="check",command=checkage)

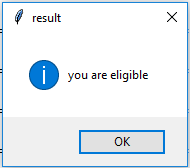
u.pack(side=LEFT)

a.pack(side=RIGHT)

b1.pack(side=BOTTOM)

root.mainloop()

**output**



**PROGRAM TO MAKE SHAPES**

**Program**

from tkinter import\*

top=Tk()

w=Canvas(top,width=300,height=200)

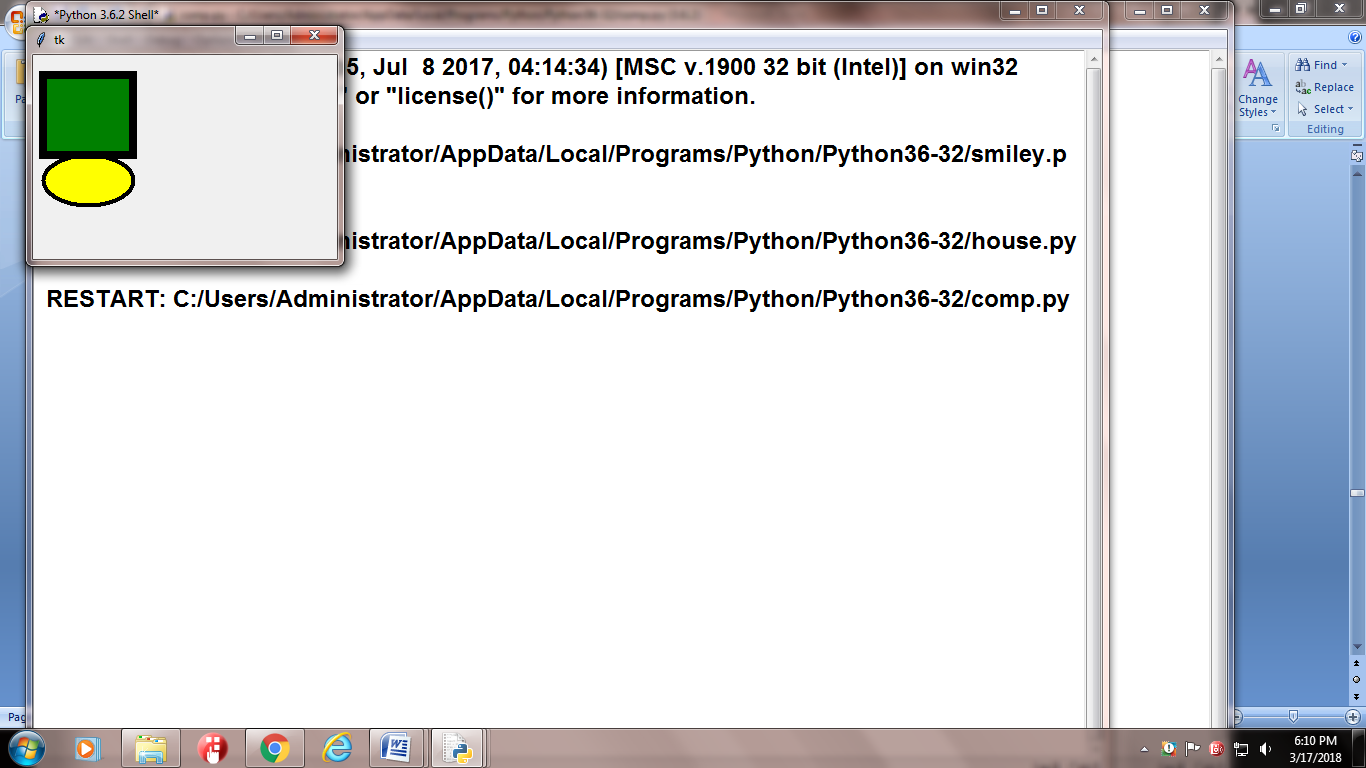
w.pack()

w.create\_oval(10,150,100,100,fill="yellow",width=4)

w.create\_rectangle(10,20,100,100,fill="green",width=8)

top.mainloop()

**output:**



**Program to draw oval**

**PROGRAM:**

from tkinter import \*

canvas\_width=190

canvas\_height=150

master=Tk()

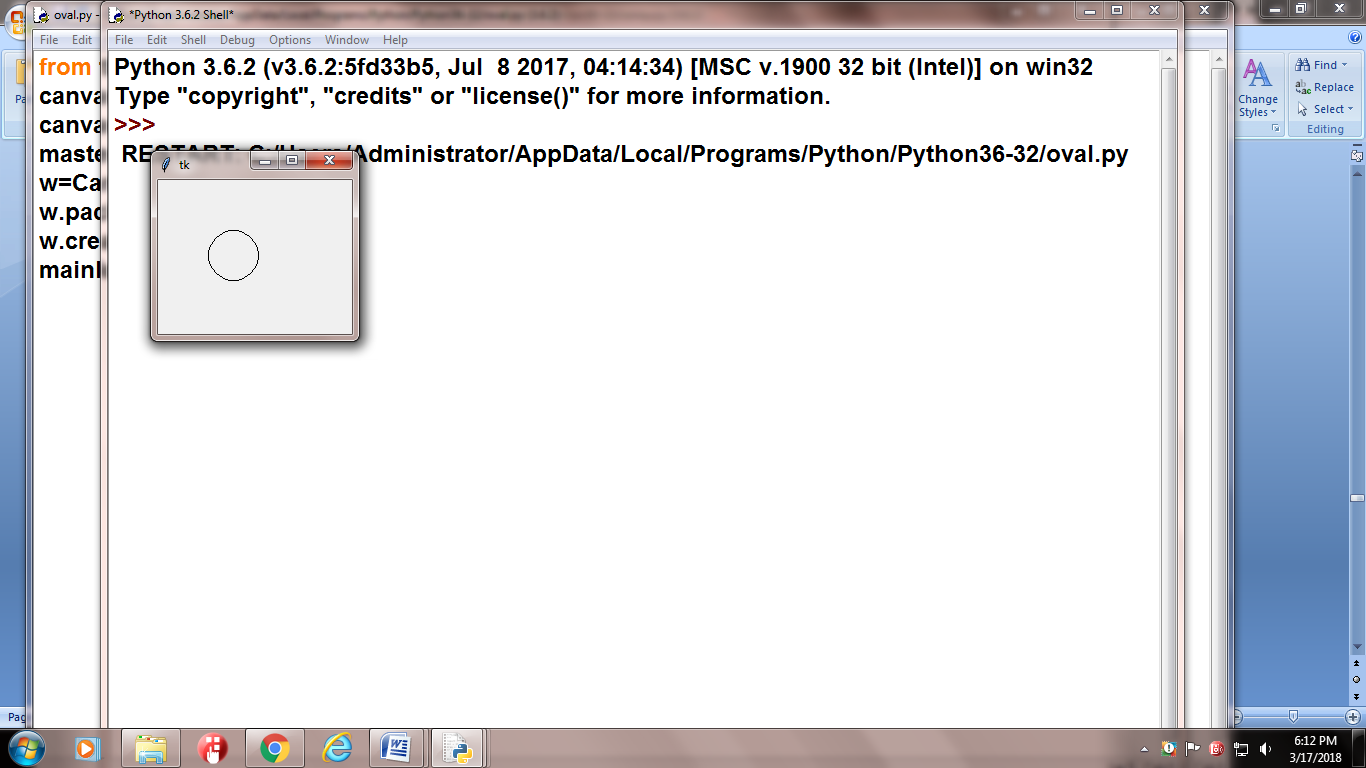
w=Canvas(master,width=canvas\_width,height=canvas\_height)

w.pack()

w.create\_oval(50,50,100,100)

mainloop()

**OUTPUT**



**Program to create line**

**PROGRAM**

from tkinter import \*

root=Tk()

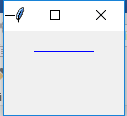
L=Canvas(root,width=60,height=80)

L.pack()

line=L.create\_line(0,20,70,20,fill='blue')

mainloop()

**output:**



**Program to draw smiley**

from tkinter import \*

root=Tk()

a=Canvas(root,width=500,height=500)

a.grid(row=0,column=0)

a.create\_oval(100,100,400,400,fill='yellow')

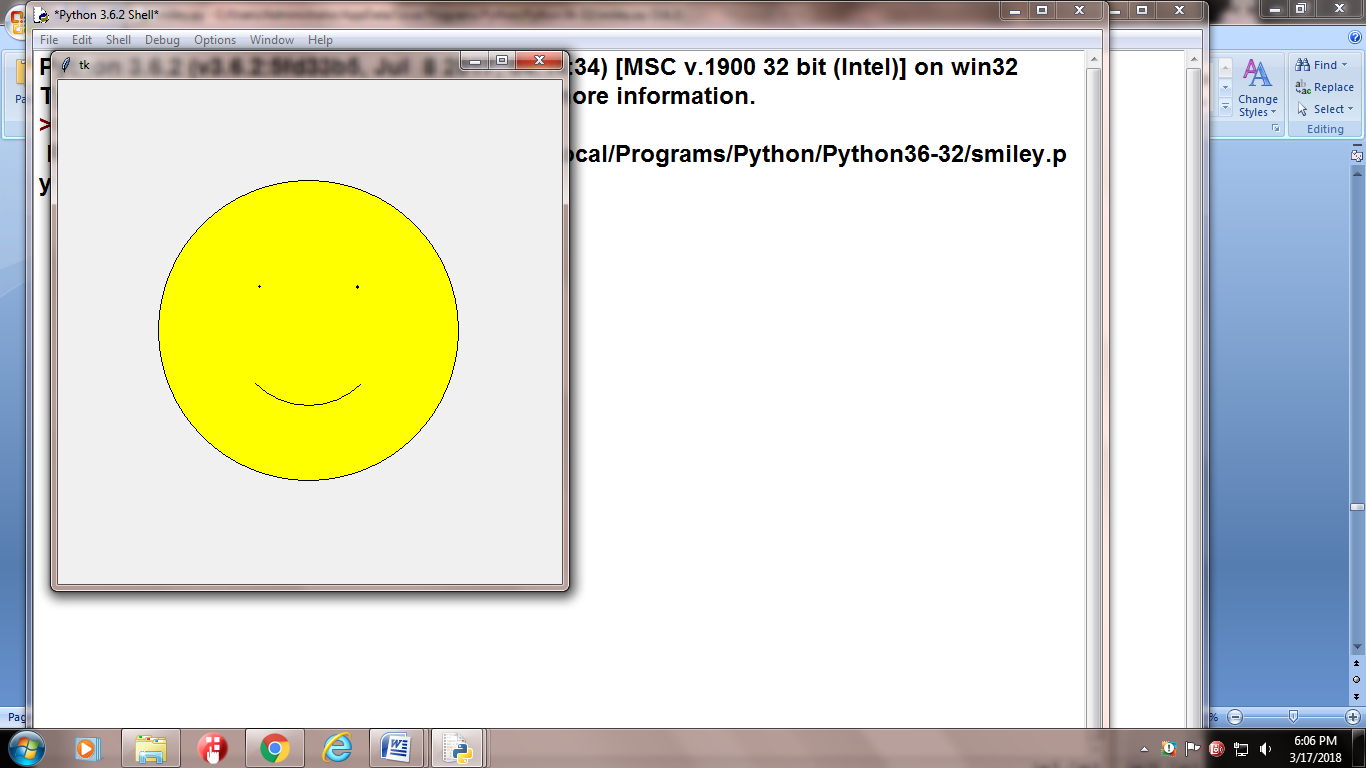
a.create\_arc(175,175,325,325,start=225,extent=90,style=ARC)

a.create\_oval(200,205,202,207,fill='black')

a.create\_oval(298,205,300,208,fill='black')

root.mainloop()

**output:**



**Program to draw house**

from tkinter import \*

root=Tk()

w=Canvas(root,width=400,height=500,bg='black')

w.pack()

w.create\_rectangle(100,200,350,300,fill='yellow')

w.create\_rectangle(200,250,250,300,fill='red')

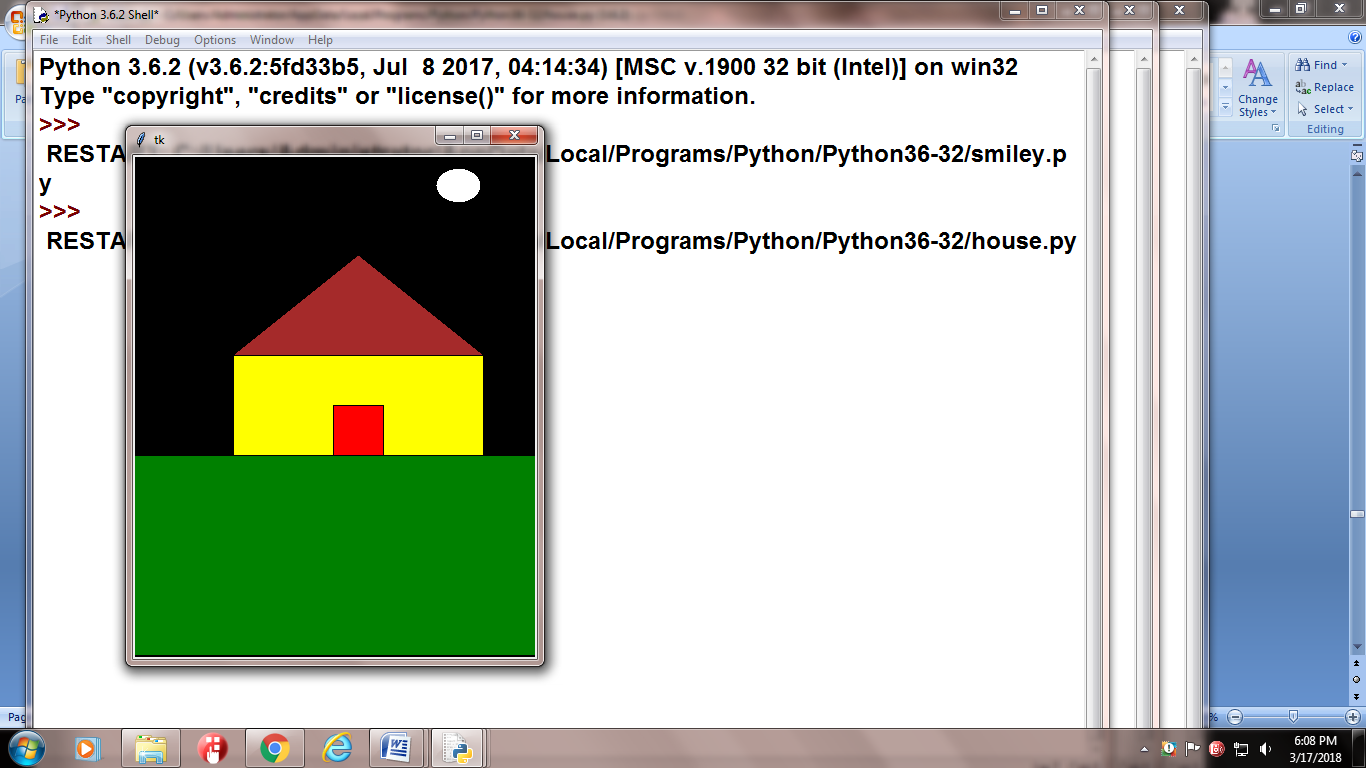
w.create\_polygon(100,200,225,100,350,200,fill='brown')

w.create\_oval(300,10,350,50,fill='white',width=6)

w.create\_rectangle(0,300,450,500,fill='green')

root.mainloop()

**output:**



**PRACTICAL 6**

**AIM:-PROGRAM TO CREATE SERVER-CLIENT AND EXCHANGE BASIC INFORMATION**

**PROGRAM:**

import socket

s=socket.socket()

print("socket successfully created")

port=12345

s.bind(('',port))

print("socket binded to %s",(port))

s.listen(5)

print("socket is listening")

while True:

c,addr=s.accept()

print("got connection from",addr)

c.send(b'thank you for getting connected')

c.close()

**output:\_**

>>>

======================== RESTART: E:\fycs14\server.py ========================

socket successfully created

socket binded to %s 12345

socket is listening

got connection from ('127.0.0.1', 50556)

>>>

import socket

s=socket.socket()

port=12345

s.connect(('127.0.0.1', port))

print (s.recv(1024))

s.close()

**output:**

>>>

======================== RESTART: E:\fycs14\client.py ========================

b'thank you for getting connected'

2]**Program to connecting to google ip address**

import socket

import sys

try:

s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

print("socket successfully created ")

except socket.error as err:

print("socket creation failed with error %s"%(err))

port=80

try:

host\_ip=socket.gethostbyname('www.google.com')

except socket.gaierror:

print("there was an error resolving the host ")

sys.exit()

s.connect((host\_ip,port))

print("the socket has successfully connected to google on port==%s"%(host\_ip))

**output:-**

================= RESTART: E:/fycs14/connection to google.py =================

socket successfully created

the socket has successfully connected to google on port==74.125.200.106

>>>

================= RESTART: E:/fycs14/connection to google.py =================

socket successfully created

the socket has successfully connected to google on port==74.125.200.104

================= RESTART: E:/fycs14/connection to google.py =================

socket successfully created

the socket has successfully connected to google on port==74.125.200.104

**PYTHON DB\_CONNECTIVITY PROGRAMS**

**TABLE CREATION AND INSERTION-**

**DB\_Create\_table.py**

**Program:**

import sqlite3

con=sqlite3.connect('population.db')

cur=con.cursor()

cur.execute("DROP TABLE IF EXISTS PopByRegion")

cur.execute('CREATE TABLE PopByRegion(Region TEXT, Population INTEGER)')

ans="yes"

count=0

while ans=="yes":

x=input("ENTER THE REGION NAME...")

y=int(input("ENTER THE POPULATION FOR %s" %x))

cur.execute("INSERT INTO PopByRegion(Region,Population) VALUES('%s',%d)" %(x,y))

count=count+1

ans=input("DO YOU WANT TO CONTINUE INSERTION...??")

print("You have inserted %d records" %count)

con.commit()

con.close()

**OUTPUT**:-

ENTER THE REGION NAME...KOLKATTA

ENTER THE POPULATION FOR KOLKATTA780000

DO YOU WANT TO CONTINUE INSERTION...??yes

ENTER THE REGION NAME...delhi

ENTER THE POPULATION FOR delhi36000

DO YOU WANT TO CONTINUE INSERTION...??no

You have inserted 2 records

**FETCHING RECORDS-**

**DB\_Reading\_records.py**

**Program:**

import sqlite3

con=sqlite3.connect('population.db')

cur=con.cursor()

sql = "SELECT \* FROM PopByRegion WHERE Population > 1000 "

try:

cur.execute(sql)

results = cur.fetchall()

for row in results:

rn = row[0]

po = row[1]

print(rn)

print(po)

except:

print ("Error: unable to fetch data")

con.close()

**OUTPUT**:-

KOLKATA

780000

delhi

36000

**TABLE UPDATION-**

**DB\_Update\_records.py**

**Program:**

import sqlite3

con=sqlite3.connect('population.db')

cur=con.cursor()

sql = "UPDATE PopByRegion SET Population=Population+50000 WHERE Region = 'dahanu' "

try:

cur.execute(sql)

print ("Record Updated...")

sql2="select \* from PopByRegion where Region = 'dahanu'"

cur=con.cursor()

cur.execute(sql2)

results = cur.fetchone()

print("region %s " %results[0])

print("Population %d" %results[1])

con.commit()

except:

con.rollback()

con.close()

**OUTPUT**:-

Record Updated...

region dahanu

Population 209000

**Program to show database connectivity**

i) Write a program to insert the following information:

Table: Item

Itemno Iname Price Quantity

101 Geometry Box 50 100

102 Soap 100 50

103 Perfume 150 25

104 Pen 50 200

105 Pencil 20 100

Write queries based upon Item table given

a) Display all items information.

b) Display item name and price value.

c) Display soap information.

d) Display the item information whose name starts with letter 'p'.

e) Display a report with item number, item name and total price. (total price = price \* quantity).

f) Display item table information in ascending order based upon item name.

g) Display item name and price in descending order based upon price.

h) Display item name, whose price is in between 50 to 100.

i) Add new column totalprice with number (10, 2).

j) Increase price value by 100.

k) Fill up totalprice = price \* quantity.

l) Remove pen information.

m) Remove totalprice column.

n) Remove whole item structure.

**Program:**

import sqlite3

con=sqlite3.connect('item.db')

cur=con.cursor()

def select():

sql = "SELECT \* FROM item"

try:

cur.execute(sql)

results = cur.fetchall()

print("ITEMNO","INAME","PRICE","QUANTITY")

for row in results:

ITEMNO = row[0]

INAME = row[1]

PRICE = row[2]

QUANTITY = row[3]

print(ITEMNO,INAME,PRICE,QUANTITY)

except:

print ("Error: unable to fetch data")

cur.execute("drop table if exists item")

cur.execute('CREATE TABLE item(ITEMNO int,INAME char(20),PRICE bigint,QUANTITY smallint)')

cur.execute("insert into item(ITEMNO,INAME,PRICE,QUANTITY)values(%d,'%s',%d,%d)"%(101,"Geometry Box",50,100))

cur.execute("insert into item(ITEMNO,INAME,PRICE,QUANTITY)values(%d,'%s',%d,%d)"%(102,"Soap",100,50))

cur.execute("insert into item(ITEMNO,INAME,PRICE,QUANTITY)values(%d,'%s',%d,%d)"%(103,"Perfume",150,25))

cur.execute("insert into item(ITEMNO,INAME,PRICE,QUANTITY)values(%d,'%s',%d,%d)"%(104,"Pen",50,200))

cur.execute("insert into item(ITEMNO,INAME,PRICE,QUANTITY)values(%d,'%s',%d,%d)"%(105,"Pencil",20,100))

select()

cur.execute("select iname,price from item")

print(cur.fetchall())

cur.execute("select \* from item where iname='Soap'")

print(cur.fetchall())

cur.execute("select \* from item where iname like'P%'")

print(cur.fetchall())

cur.execute("select iname,itemno,(price\*quantity)as total from item")

print(cur.fetchall())

cur.execute("select iname from item order by iname")

print(cur.fetchall())

cur.execute("select iname,price from item order by price desc")

print(cur.fetchall())

cur.execute("select iname from item where price between 50 and 100")

print(cur.fetchall())

cur.execute("alter table item add totalprice decimal(10,2)")

cur.execute("update item set price=price+100")

print(cur.fetchall())

cur.execute("select price from item")

print(cur.fetchall())

cur.execute("update item set totalprice=(price\*quantity)")

cur.execute("select \* from item")

print(cur.fetchall())

cur.execute("delete from item where iname='Pen'")

cur.execute("select \* from item")

print(cur.fetchall())

cur.execute("drop table item")

cur.execute("select \* from item")

con.commit()

con.close()

**output:**

ITEMNO INAME PRICE QUANTITY

101 Geometry Box 50 100

102 Soap 100 50

103 Perfume 150 25

104 Pen 50 200

105 Pencil 20 100

[('Geometry Box', 50), ('Soap', 100), ('Perfume', 150), ('Pen', 50), ('Pencil', 20)]

[(102, 'Soap', 100, 50)]

[(103, 'Perfume', 150, 25), (104, 'Pen', 50, 200), (105, 'Pencil', 20, 100)]

[('Geometry Box', 101, 5000), ('Soap', 102, 5000), ('Perfume', 103, 3750), ('Pen', 104, 10000), ('Pencil', 105, 2000)]

[('Geometry Box',), ('Pen',), ('Pencil',), ('Perfume',), ('Soap',)]

[('Perfume', 150), ('Soap', 100), ('Geometry Box', 50), ('Pen', 50), ('Pencil', 20)]

[('Geometry Box',), ('Soap',), ('Pen',)]

[]

[(150,), (200,), (250,), (150,), (120,)]

[(101, 'Geometry Box', 150, 100, 15000), (102, 'Soap', 200, 50, 10000), (103, 'Perfume', 250, 25, 6250), (104, 'Pen', 150, 200, 30000), (105, 'Pencil', 120, 100, 12000)]

[(101, 'Geometry Box', 150, 100, 15000), (102, 'Soap', 200, 50, 10000), (103, 'Perfume', 250, 25, 6250), (105, 'Pencil', 120, 100, 12000)]

**PRACTICAL 7**

**AIM:-PROGRAM TO SEND EMAIL & READ CONTENTS OF URL**

**Program:**

import smtplib

TO="rashmitha.kotian10@gmail.com"

SUBJECT="TEST python MAIL"

TEXT="HERE IS A MESSAGE FROM PYTHON"

#GMAIL SIGNIN

gmail\_sender="rashmitha.kotian99@gmail.com"

gmail\_passwd="\*\*\*\*\*\*\*\*"

server = smtplib.SMTP("smtp.gmail.com",587)

server.ehlo()

server.starttls()

server.login(gmail\_sender,gmail\_passwd)

BODY='\r\n'.join(['To:%s' % TO,

'From:%s' %gmail\_sender,

'Subject: %s'%SUBJECT,

'',TEXT])

try:

server.sendmail(gmail\_sender,[TO],BODY)

print('email sent')

except:

print('error sending mail')

server.quit()

**output:-**

============= RESTART: C:\Users\Administrator\Downloads\gmail.py =============

email sent

>>>

**Program to read contents of url**

**Program:**

from urllib.request import urlopen

response=urlopen('https://www.python.org/')

print("response:",response)

print("url is:",response.geturl())

print("this gets the code:",response.code)

print("the headers are:",response.info())

print("\n the date is :",response.info()['date'])

print("\n the server is :",response.info()['server'])

html=response.read()

print("get all data:",html)

print("get length:",len(html))

**output:**

response: <http.client.HTTPResponse object at 0x05E7C0F0>

url is: https://www.python.org/

this gets the code: 200

the headers are: Server: nginx

Content-Type: text/html; charset=utf-8

X-Frame-Options: SAMEORIGIN

x-xss-protection: 1; mode=block

X-Clacks-Overhead: GNU Terry Pratchett

Via: 1.1 varnish

Fastly-Debug-Digest: a63ab819df3b185a89db37a59e39f0dd85cf8ee71f54bbb42fae41670ae56fd2

Content-Length: 48844

Accept-Ranges: bytes

Date: Tue, 13 Mar 2018 06:28:44 GMT

Via: 1.1 varnish

Age: 2915

Connection: close

X-Served-By: cache-iad2123-IAD, cache-sea1031-SEA

X-Cache: HIT, HIT

X-Cache-Hits: 3, 29

X-Timer: S1520922525.746005,VS0,VE0

Vary: Cookie

Strict-Transport-Security: max-age=63072000; includeSubDomains

the date is : Tue, 13 Mar 2018 06:28:44 GMT

the server is : nginx

get all data: b'<!doctype html>\n<!--[if lt IE 7]> <html class="no-js ie6 lt-ie7 lt-ie8 lt-ie9"> <![endif]-->\n<!--[if IE 7]> <html class="no-js ie7 lt-ie8 lt-ie9"> <![endif]-->\n<!--[if IE 8]> <html class="no-js ie8 lt-ie9"> <![endif]-->\n<!--[if gt IE 8]><!--><html class="no-js" lang="en" dir="ltr"> <!--<![endif]-->\n\n<head>\n <meta charset="utf-8">\n <meta http-equiv="X-UA-Compatible" content="IE=edge">\n\n <link rel="prefetch" href="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js">\n\n <meta name="application-name" content="Python.org">\n <meta name="msapplication-tooltip" content="The official home of the Python Programming Language">\n <meta name="apple-mobile-web-app-title" content="Python.org">\n <meta name="apple-mobile-web-app-capable" content="yes">\n <meta name="apple-mobile-web-app-status-bar-style" content="black">\n\n <meta name="viewport" content="width=device-width, initial-scale=1.0">\n

get length: 13636