Experiment number 1: Basics of Python

"""lambda, map, reduce, filter"""

"""A lambda function is a small anonymous function.
A lambda function can take any number of arguments, but can only have one expression.
no need to define a function, pass parameters, return result"""

```
x = lambda a : a + 10
print(x(5))
x = lambda a, b : a * b
print(x(5, 6))
x = lambda a, b, c : a + b + c
print(x(5, 6, 2))
"""output
15
30
13
"""
```

"""The filter() function in Python takes in a function and a list as arguments.
This offers an elegant way to filter out all the elements of a sequence “sequence”,
for which the function returns True. """

```
li = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]
odd_list = list(filter(lambda x: (x%2 != 0) , li))
print(odd_list)
even_list = list(filter(lambda x: (x%2 == 0) , li))
print(even_list)
"""output
[5, 7, 97, 77, 23, 73, 61]
[22, 54, 62]
"""
```

"""map maps element of list to another value as per expression in lambda"""

```
square_list = list(map(lambda x: x*x,li))
print(square_list)
cube_list = list(map(lambda x:x*x*x,li))
print(cube_list)
"""output
[25, 49, 484, 9409, 2916, 3844, 5929, 529, 5329, 3721]
[125, 343, 10648, 912673, 157464, 238328, 456533, 12167, 389017, 226981]
"""
```

"""reduces a list to result"""

```
from functools import reduce
li = [5, 8, 10, 20, 50, 100]
sum = reduce((lambda x, y: x + y), li)
print(sum)
mult = reduce((lambda x,y:x*y),li)
print(mult)
"""output
193
40000000
"""
```

```
li = list(range(1,11))
sum_cube_even = reduce( lambda x,y: x+y, list( map( lambda x:x*x*x, list( filter( lambda x: x%2==0, li )
) ) ) )
print(sum_cube_even)
"""output
1800
"""
```

""" Demonstration of string function.
WAP to demonstrate 10 function of string. """

```
str = "a  abcAd  "
print(str.capitalize())
print(str.islower())
print(str.isupper())
print(str.isdigit())
print(str.isalpha())
print(str.lower())
print(str.upper())
print(str.strip())
print(str.title())
print(str.replace("Ad","Cd"))
"""output
A  abcad
False
False
False
False
a  abcad
A  ABCAD
a  abcAd
A  Abcad
a  abcCd
"""
```

"""WAP to sort a string in alphabetical order. """

```
str = input("enter string:")
words = str.split()
words.sort()
for word in words:
   print(word,end=" ")
"""output
enter string:this is string
is string this
"""
```

"""WAP to seacrh a string in given sentence. """

```
str = input("enter string:")
search_str = input("search string:")
print(str.find(search_str))
"""output
enter string:this is string
```

```
search string:this
0
"""
```

"""WAP to substitute a sub string with a given string. """

```
str = input("enter string:")
search_str = input("search string:")
rep_str = input("replace string:")
print(str.replace(search_str,rep_str))
"""output
enter string:this is string
search string:is
replace string:was
thwas was string
"""
```

"""WAP to count number of vowels in a given string. """

```
string=input("Enter string:")
vowels=0
for i in string:
    if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U'):
        vowels=vowels+1
print("Number of vowels are:")
print(vowels)
"""output
Enter string:this is string
Number of vowels are:
3
"""
```

"""WAP implement encryption algorithm(cipher). """

```
def enc(s,k):
  result=""
  for i in range(len(s)):
    char = s[i]
    if(char.isupper()):
      result += chr((ord(char) + k-65) % 26 + 65)
    else:
      result += chr((ord(char) + k - 97) % 26 + 97)
  return result

string=input("Enter string:")
key = int(input("Enter key:"))
print("cipher text",enc(string,key))
"""output
Enter string:this is string
Enter key:3
cipher text wklvqlvqvwulqj
"""
```

```
"""WAP to capitalize the first character of each word. """
str = input("enter string:")
print(str.title())
"""output
enter string:this is string
This Is String
"""

"""WAP to count number of spaces. """
string=input("Enter string:")
print(string.count(' '))
"""output
Enter string:this is string
2
"""

"""WAP to copy a string without spaces. """
str=input("Enter string:")
str2 = str.replace(" ","")
print(str2)
"""output
Enter string:this is string
thisisstring
"""

"""WAP to count number of characters in a string. """
str=input("Enter string:")
print(len(str))
"""output
Enter string:this is string
14
"""

"""Demonstration of list function.
WAP to create 5 list using comprehension method. """
x = [i for i in range(10)]
print(x)

x = [i*i for i in range(10)]
print(x)

x = [i*i*i for i in range(10)]
print(x)

listOfWords = ["this","is","a","list","of","words"]
items = [ word[0] for word in listOfWords ]
print(items)

string = "Hello 12345 World"
numbers = [x for x in string if x.isdigit()]
print(numbers)
```

```
"""output
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
[0, 1, 8, 27, 64, 125, 216, 343, 512, 729]
['t', 'i', 'a', 'l', 'o', 'w']
['1', '2', '3', '4', '5']
"""

"""WAP to add 2 arrays using list. """
l1 = [1,2,3,4]
l2 = [5,6,7,8]
l3 = []
for i in range (0,len(l1)):
   l3.append(l1[i]+l2[i])
print(l3)
"""output
[6, 8, 10, 12]
"""

"""WAP to demonstrate all list functions. """
l1 = []
l1.append("1")
l2 = ['a','b','a']
l1.extend(l2)
print(l1)
l1.insert(0,'A')
print(l1)
l1.remove('A')
print(l1)
print(l1.index('a'))
print(l1.count('a'))
l1.pop()
print(l1)
l1.reverse()
print(l1)
l1.sort()
print(l1)
l1.clear()
print(l1)
print(len(l1))
print(max(l2))
print(min(l2))
"""output
['1', 'a', 'b', 'a']
['A', '1', 'a', 'b', 'a']
['1', 'a', 'b', 'a']
1
2
['1', 'a', 'b']
['b', 'a', '1']
['1', 'a', 'b']
[]
```

```
0
b
a
"""
```

"""WAP to implement stack using list. """

```
stack = []
stack.append('a')
stack.append('b')
print(stack)
stack.pop()
print(stack)
"""output
['a', 'b']
['a']
"""
```

"""WAP to perform linear search in an array"""

```
list = [1,3,4,2,5]
key = 3
h=0
for i in list:
   if i==key:
      h=1
if h==0:
   print("Not found")
else:
   print("Found")
"""output
Found
"""
```

"""Demonstration of tuple, dictionary and set."""

"""a. WAP a program to create a tuple having 20 numbers."""

```
t = ()
for i in range(0,21):
   t = t + (i,)
print(t)
"""(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)"""
```

"""b. WAP to create a tuple of 20 user entered numbers and create another tuple of even numbers from the before tuple."""

```
t_even = ()
for i in t:
   if i%2==0:
      t_even = t_even + (i,)
print(t_even)
"""(0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20)"""
```

"""c. WAP to demonstrate any 10 tuple methods."""

```
print(t.count(2))  # 1
print(t.index(4))  # 4
print(3 in t)  # True
for name in ('John','Kate'):
    print("Hello",name)
"""Hello John
Hello Kate"""
print(all(t))  # False
print(len(t))  # 21
print(sorted(t)) # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
print(sum(t))  # 210
```

"""d. WAP to check if the given number is present in the tuple or not."""

```
print(3 in t)  # False
```

"""e. WAP to t1==t2, max(), min() methods."""

```
if t==t_even:
   print("Equal")
else:
   print("Not Equal")  # Not Equal
print(max(t))  # 20
print(min(t))  # 0
```

"""f. WAP to create a tuple of colours, tuple of fruits, tuple of quantity. Create a list using ZIP for colour and fruit, then create a tuple of fruit and quantity, and create a zip of fruit, color and quantity."""

```
colors = ('red','yellow','green')
fruits = ('apple','mango','guava')
quantity = (10,20,30)
cf = zip(colors,fruits)
cflist = list(cf)
print(cflist)  # [('red', 'apple'), ('yellow', 'mango'), ('green', 'guava')]
fq = zip(fruits,quantity)
fqtuple = tuple(fq)
print(fqtuple)  # (('apple', 10), ('mango', 20), ('guava', 30))
fcq = zip(fruits,colors,quantity)
fcqlist = list(fcq)
print(fcqlist)  # [('apple', 'red', 10), ('mango', 'yellow', 20), ('guava', 'green', 30)]
```

"""g. WAP to create a dictionary with any 5 key value pairs, print the dictionary, add an element, and delete and element."""

```
dict = {
   "name":"ABC",
   "age":23,
   "rollno":123,
   "per":88.85,
   "pr":1
}
print(dict)  # {'name': 'ABC', 'age': 23, 'rollno': 123, 'per': 88.85, 'pr': 1}
dict["class"] = "SE"
print(dict)  # {'name': 'ABC', 'age': 23, 'rollno': 123, 'per': 88.85, 'pr': 1, 'class': 'SE'}
dict.pop("pr")
print(dict)  # {'name': 'ABC', 'age': 23, 'rollno': 123, 'per': 88.85, 'class': 'SE'}
```

```
"""h. WAP to create 2 dictionary using comprehnesive method."""
i = [1,2,3,4,5]
d1 = {}
for x in i:
    d1[x] = x*x
print(d1)  # {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
fruits = ['apple', 'mango', 'banana','cherry']
d2 = {}
for x in fruits:
    d2[x] = len(x)
print(d2)  # {'apple': 5, 'mango': 5, 'banana': 6, 'cherry': 6}

"""i. WAP to demonstrate 5 dictionary methods."""
print(d2)  # {'apple': 5, 'mango': 5, 'banana': 6, 'cherry': 6}
d2.clear()
print(d2)  # {}
print(d2.get('apple',"None"))  # None
print(d1.items())  # dict_items([(1, 1), (2, 4), (3, 9), (4, 16), (5, 25)])
d1.pop(2)
print(d1)  # {1: 1, 3: 9, 4: 16, 5: 25}
print(d1.values())  # dict_values([1, 9, 16, 25])
```

Experiment number 2: Decision making and functions

```
def prime(x):
   if(x<3):
      print("Not Prime")
      return
   else:
      for j in range(2,i-1):
         if(i%j==0):
            print("Not Prime")
            return
   print("Prime")
   return

def fib(x):
   a=0
   b=1
   print(a,end=", ")
   x-=1
   while(x!=0):
      c = a+b
      a=b
      b=c
      print(a,end=", ")
      x-=1

def gcd(a,b):
   if b==0:
      return a
   else:
      return gcd(b,a%b)

def cal(a,b):
   print("a+b:",a+b)
   print("a-b:",a-b)
   print("a*b:",a*b)
   if b!=0:
      print("a/b:",a/b)
   else:
      print("a/b:error")
```

"""WAP to calculate area of circle"""

```
r = float(input("Enter radius:"))
print("area of cirlce:",3.14*r*r)
```

"""**output**

```
Enter radius:5
area of cirlce: 78.5
"""
```

"""WAP to check prime number"""

```
i = int(input("enter number:"))
prime(i)
```

```
"""output
enter number:13
Prime
"""
```

"""WAP to generate fibonacci serires"""

```
i = int(input("Enter number:"))
print("Fibonacci series: ",end="")
fib(i)
"""output
Enter number:5
Fibonacci series: 0, 1, 1, 2, 3,
"""
```

"""WAP to calculate GCD. """

```
i = int(input("First number:"))
j = int(input("Second number:"))
print(gcd(i,j))
"""output
First number:64
Second number:24
8
"""
```

"""WAP to create a simple calculator that can add, multiply, subtract, using functions. """

```
i = int(input("a:"))
j = int(input("b:"))
cal(i,j)
"""output
a:10
b:2
a+b: 12
a-b: 8
a*b: 20
a/b: 5.0
"""
```

"""WAP to generate pattern

```
   *
   **
   ***
   ****"""
n=int(input("No of lines:"))
for i in range(0,n+1):
   for j  in range(0,i):
      print("*",end=" ")
   print("")
```

```
"""output
No of lines:5

*
* *
* * *
* * * *
* * * * *
"""

"""WAP to generate pattern
   1
   2 3
   4 5 6
   7 8 9 10"""
n=int(input("No of lines:"))
x=1
for i in range(0,n+1):
   for j  in range(0,i):
      print(x,end=" ")
      x+=1
   print("")
"""output
No of lines:5

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
"""

"""WAP to generate pattern
   A
   A B
   A B C
   A B C D"""
n=int(input("No of lines:"))
for i in range(0,n+1):
   x='A'
   for j  in range(0,i):
      print(x,end=" ")
      x = chr(ord(x)+1)
   print("")
"""output
No of lines:4

A
A B
A B C
A B C D
"""
```

```
"""WAP to generate pattern
       *
      **
     ***
    ****"""
n=int(input("No of lines:"))
for i in range(0,n+1):
    for j in range(0,n-i):
        print(" ",end=" ")
    for j  in range(0,i):
        print("*",end=" ")
    print("")
"""output
        *
      * *
    * * *
  * * * *
* * * * *
"""
```

```
"""WAP to print even numbers between m and n. """
m = int(input("m:"))
n = int(input("n:"))
for i in range(m+1,n):
    if i%2==0:
        print(i,end=", ")
"""output
m:10
n:20
12, 14, 16, 18,
"""
```

Experiment number 3: Object oriented programming using python

```
"""Object oriented programming.
a. class and object.
b. constructors.
c. class variables/ class methods.
d. inheriatnce.
e. super.
f. polymorphism.
g. constructor with inheritance. """

class MyClass:
   var = "blah"
   def __init__(self,x="blank"):
      self.var=x
   def func(self):
      print("This is a message inside the class.")
      print("var = ",self.var)

obj = MyClass()
print(obj.var)
obj.func()
obj2 = MyClass()
obj2.var = "ABC"
print(obj2.var)
obj2.func()
obj3 = MyClass("third")
obj3.func()

class Polygon:
   sides=0
   def __init__(self,n=0):
      self.sides=n
   def count(self):
      print("no of sides:",self.sides)

class Triangle(Polygon):
   a=0
   b=0
   c=0
   def __init__(self,x,y,z=None):
      super().__init__(3)
      if z is None:
         self.a = x
         self.b = y
      else:
         self.a = x
         self.b = y
         self.c = z

   def count(self):
      print("no of sides of triangle:",self.sides)
   def print_area(self):
        print("Area",(self.a*self.b)/2)

t = Triangle(10,20)
t.count()
t.print_area()

"""
```

Output

```
blank
This is a message inside the class.
var =  blank
ABC
This is a message inside the class.
var =  ABC
This is a message inside the class.
var =  third
no of sides of triangle: 3
Area 100.0"""
```

Experiment number 4: File handling

```
"""To implement file handling
a. WAP to write into file.
b. WAP to read into the file.
c. WAP to count the no. of words in a file.
d. WAP to count the no. spaces in a file.
e. WAP to copy content in uppercase from file 1 to file 2
f. WAP to demonstrate any 10 file methods.
g. WAP to demonstrate use of 'seek' and 'tell' methods.
h. WAP to demonstrate zipping and unzipping.
j. WAP to copy an image file. """

f = open('temp.txt','w')
f.write("This is first line\n")
f.write("This is second line")
f.close()

f = open('temp.txt','r')
print(f.read())
f.close()
"""output
This is first line
This is second line
"""

n=0
s=0
with open("temp.txt",'r') as file, open("temp2.txt",'w') as file2:
    data = file.readlines()
    for line in data:
        word = line.split()
        x = line.upper()
        file2.write(x)
        for w in word:
            n+=1
            s+=1
        s-=1
print(n)
print(s)
"""output
8
6
"""

f = open("temp2.txt",'r')
print(f.read())
print(f.tell())
f.seek(32)
print(f.readline())
f.close()
```

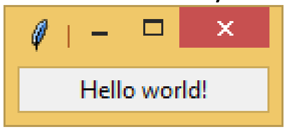
```
"""output
THIS IS FIRST LINE
THIS IS SECOND LINE
39
ND LINE
"""

import zipfile
import os
zfile = zipfile.ZipFile('zfile.zip','w')
zfile.write('pyCal.py', compress_type=zipfile.ZIP_DEFLATED)
zfile.close()
"""output
Zip file created, check folder
"""

zfile = zipfile.ZipFile('zipDrive.zip','w')
for folder, subfolder, files in os.walk('./'):
    for file in files:
        if file.endswith('.py'):
            zfile.write(file,compress_type=zipfile.ZIP_DEFLATED)
zfile.close()
"""output
Zip file created, check folder
"""

import shutil
shutil.copy('icon.gif','icon2.gif')
"""output
Image file copied, check folder
"""
```

Experiment number 5: GUI programming and database

"""GUI programming in Python. """

"""Basic tkinter GUI"""

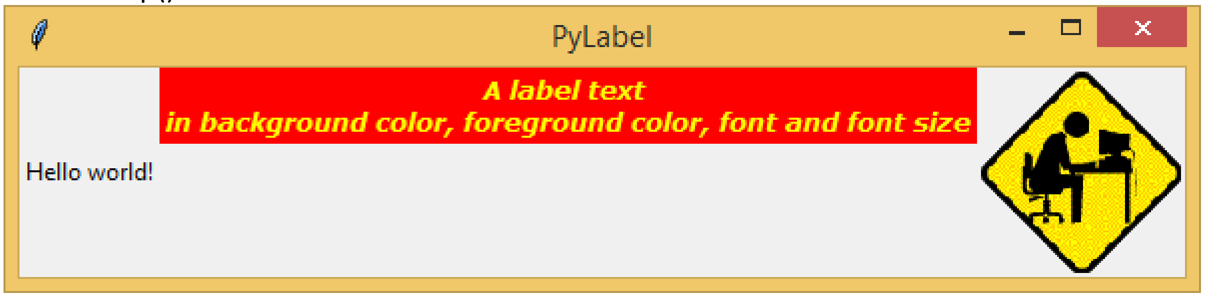
```
import tkinter as tk
"""importing tkinter module if python 3.X else Tkinter if python 2.X"""
m = tk.Tk()
m.title('PyGui')
"""creating m as main window to which all other widgets will be attached
title of window = PyGui"""
l1 = tk.Label(m, text="Hello world!")
"""label widget with master window as root window and text to display is Hello world!"""
l1.pack()
"""attched and fit the widget to master window"""
m.mainloop()
"""main loop which is an infinite loop until window is closed, waits for events on widgets after opening main window
there can be only one mainloop of main window"""
```



Hello world!

"""tkinter Label"""

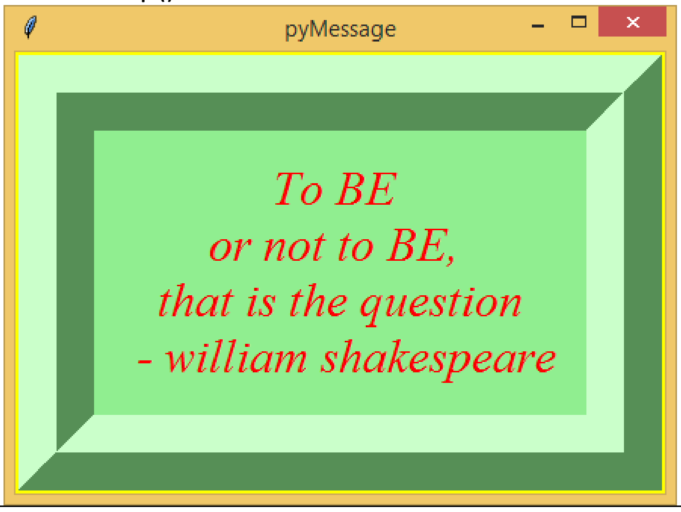
```
import tkinter as tk
m = tk.Tk()
m.title('PyLabel')

l1 = tk.Label(m, text="Hello world!")
"""label widget user can view but can not interact with"""
"""label widget with master window as root window and text to display is Hello world!"""
l1.pack(side="left")
"""attched as left align and fit the widget to master window"""

"""image as label"""
logo = tk.PhotoImage(file="icon.gif")
l2 = tk.Label(m, image=logo)
l2.pack(side="right")

caption = "A label text \nin background color, foreground color, font and font size"
l3 = tk.Label(m,text=caption,bg="red",fg="yellow", font="Verdana 10 bold italic", justify="center")
l3.pack()

m.mainloop()
```



PyLabel

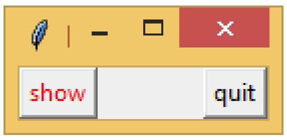
A label text
in background color, foreground color, font and font size

Hello world!

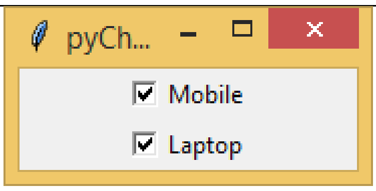
```
"""tkinter Message"""
"""Message widgete can be used to display short text message
similar to Label but much more flexible"""
import tkinter as tk
m = tk.Tk()
m.title('pyMessage')
messageText = "To BE \nor not to BE, \nthat is the question\n - william shakespeare"
msg = tk.Message(m,text=messageText)
msg.config(bg="light green",
        bd="50",
        cursor="spider",
        font=("times",24,"italic"),
        fg="red",
        justify="center",
        padx=20,
        pady=20,
        relief="ridge",
        takefocus=True,
        highlightbackground="yellow",
        highlightthickness=2,
        highlightcolor="red"
        )
"""bg or background for background color
bd or borderwidth
cursor for type of cursor to show when hovered on message widget
font to change font name, size, style
fg or foreground for foreground color
justify text left, right, center
padx horizontal padding
pady vertical padding
default border is flat other values sunken, raised, groove, ridge
takefocus allows focus default false, press tab key
highlightbackground color when not focus
highlightcolor color when focused
highlightthickness thickness of focus
"""
msg.pack()
m.minsize(200,200)
m.mainloop()
```


To BE
or not to BE,
that is the question
- william shakespeare

```
"""tkinter Button"""
"""a button is a widget with which user can interact with
only one font for button text
python function associated with button"""
import tkinter as tk
def show_msg():
    print("Hi")

m = tk.Tk()
m.title("PyButton")
b1 = tk.Button(m,text="show",fg="red",command=show_msg)
b1.pack(side="left")
b2 = tk.Button(m,text="quit",command=quit)
b2.pack(side="right")
m.mainloop()
```

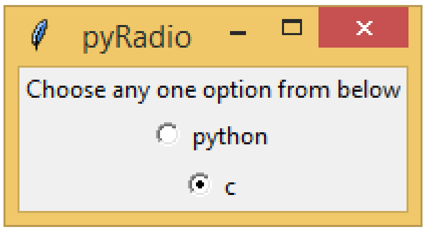


```
"""tkinter Checkbox"""
"""checkbox button or option button
choose multiple from options
if a radio button is pressed py function can be called
they are associated with same variable
"""
import tkinter as tk
def show_choice_mob():
    if(mob.get()==1):
        print("you have mobile")
    else:
        print("you do not have mobile")

def show_choice_lap():
    if(laptop.get()==1):
        print("you have laptop")
    else:
        print("you do not have laptop")

m = tk.Tk()
m.title("pyCheckbox")

mob = tk.IntVar()
cm = tk.Checkbutton(m,text="Mobile",variable=mob,width="20",command=show_choice_mob).pack()

laptop = tk.IntVar()
cl = tk.Checkbutton(m,text="Laptop",variable=laptop,width="20",command=show_choice_lap).pack()

m.mainloop()
```

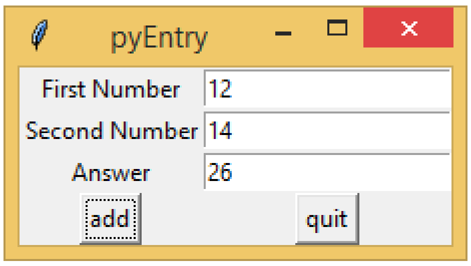


"""tkinter Radio"""

```
"""radio button or option button
choose only one from options
if a radio button is pressed py function can be called
they are associated with same variable
"""
import tkinter as tk
def show_choice():
    print(v.get())
m = tk.Tk()
m.title("pyRadio")

v = tk.IntVar()
v.set(1)  # set default value
"""similar to IntVar(), StringVar() for string, DoubleVar() for float, BooleanVar() for boolean 1 and 0"""

l = tk.Label(m,text="Choose any one option from below")
l.config(justify="center")
l.pack()

r1 = tk.Radiobutton(m,padx=10,width=20,text="python",variable=v,value=1,command=show_choice)
r1.pack()

r2 =
tk.Radiobutton(m,padx=10,width=20,text="c",variable=v,value=2,indicatoron=1,command=show_choi
ce)
"""indicatoron=0 no radio button but a button form"""
r2.pack()

m.mainloop()
```



```
"""tkinter application"""
"""Entry widget reads a string"""
import tkinter as tk

def show_add():
    a = int(e1.get(),10)  # read value of first entry, convert to int
    b = int(e2.get(),10)  # read value of second entry, convert to int
    c = a+b  # add
    e3.delete(0,tk.END)  # empty previous content of answer entry
    e3.insert(0,str(c))  # write answer in string form

m = tk.Tk()
m.title("pyEntry")

l1 = tk.Label(m,text="First Number")
l2 = tk.Label(m,text="Second Number")
l3 = tk.Label(m,text="Answer")
l1.grid(row=0,column=0)
l2.grid(row=1,column=0)
l3.grid(row=2,column=0)

e1 = tk.Entry(m)
e1.insert(0,"0")
e2 = tk.Entry(m)
e2.insert(0,"0")
e3 = tk.Entry(m)
e3.insert(0,"0")
e1.focus_set()
e1.grid(row=0,column=1)
e2.grid(row=1,column=1)
e3.grid(row=2,column=1)


b1 = tk.Button(m,text="add",command=show_add)
b2 = tk.Button(m,text="quit",command=quit)
b1.grid(row=3,column=0)
b2.grid(row=3,column=1)

m.mainloop()
```


pyEntry

First Number 12

Second Number 14

Answer 26

```
"""Database connection using Python.
Employee table
1.
a. WAP to create a database.
b. WAP to open a database.
c. WAP to add a data in database.
d. WAP to implement cursor.
e. WAP to update database.
f. WAP to delete data from database. """


import sqlite3 as sq
from sqlite3 import Error
try:
    conn = sq.connect('test2.db')
    print("Database opened")
    q = "create table if not exists std(sroll int primary key not null, sname text not null, sage int not null
)"
    try:
        c = conn.cursor()
        c.execute(q)
        conn.commit()
        print("table created")
        q1 = "insert or ignore into std(sroll, sname, sage) values(123,'ABC',23)"
        q2 = "insert or ignore into std(sroll, sname, sage) values(234,'DEF',24)"
        q3 = "insert or ignore into std(sroll, sname, sage) values(345,'XYZ',25)"
        try:
            c.execute(q1)
            c.execute(q2)
            c.execute(q3)
            conn.commit()
            print(c.lastrowid," rows inserted")
            try:
                q = "select sroll, sname from std where sage>?"
                x=23
                c.execute(q,(x,))
                conn.commit()
                rows = c.fetchall()
                for row in rows:
                    print(row)
            except Error as e:
                print(e)
            try:
                q = "update std set sage=? where sroll=?"
                x=25
                y=123
                c.execute(q,(x,y))
                conn.commit()
                print(c.rowcount," rows updated")
            except Error as e:
                print(e)
            try:
```

```
            q = "delete from std where sage=?"
            x=25
            c.execute(q,(x,))
            conn.commit()
            print(c.rowcount," rows deleted")
         except Error as e:
            print(e)
      except Error as e:
         print(e)
   except Error as e:
      print(e)
except Error as e:
   print(e)
finally:
   conn.close()

"""
```

Output

```
Database opened
table created
4  rows inserted
(234, 'DEF')
(345, 'XYZ')
1  rows updated
2  rows deleted"""
```

Experiment number 6: Web programming

"""Program to perform UDP connection. """

"""UdpServer.py"""

```
import socket

localIP = "127.0.0.1"
localPort = 20001
bufferSize = 1024

msgFromServer = "Hello UDP Client"
bytesToSend = str.encode(msgFromServer)

# Create a datagram socket
UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)

# Bind to address and ip
UDPServerSocket.bind((localIP, localPort))

print("UDP server up and listening")

# Listen for incoming datagrams
while (True):
   bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)
   message = bytesAddressPair[0]
   clientMsg = "Message from Client:{}".format(message)
   print(clientMsg)

   address = bytesAddressPair[1]
   clientIP = "Client IP Address:{}".format(address)
   print(clientIP)

   # Sending a reply to client
   UDPServerSocket.sendto(bytesToSend, address)
```

""UdpClient.py"""

```
import socket

msgFromClient = "Hello UDP Server"
bytesToSend = str.encode(msgFromClient)

serverAddressPort = ("127.0.0.1", 20001)
bufferSize = 1024

# Create a UDP socket at client side
UDPClientSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)

# Send to server using created UDP socket
UDPClientSocket.sendto(bytesToSend, serverAddressPort)
msgFromServer = UDPClientSocket.recvfrom(bufferSize)
msg = "Message from Server {}".format(msgFromServer[0])
print(msg)
```

```
"""output
>>>python UdpServer.py
UDP server up and listening
Message from Client:b'Hello UDP Server'
Client IP Address:('127.0.0.1', 57223)

>>>python UdpClient.py
Message from Server b'Hello UDP Client'"""
```

```
"""Program to demonstrate simple socket programming. """
# connect to google using socket
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))

# default port for socket
port = 80
try:
    host_ip = socket.gethostbyname('www.google.com')
except socket.gaierror:
    print("there was an error resolving the host")

# connecting to the server
s.connect((host_ip, port))

print("the socket has successfully connected to google on port == %s" % (host_ip))
"""output
Socket successfully created
the socket has successfully connected to google on port == 172.217.160.196
"""
```

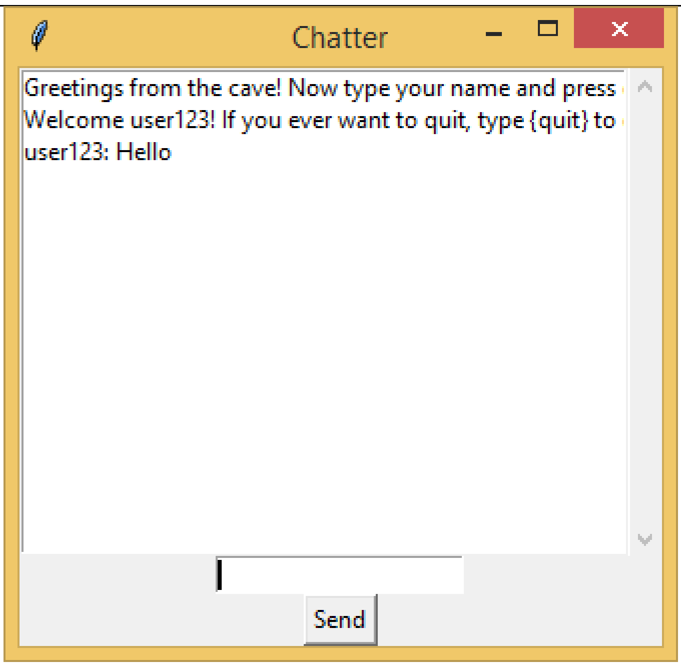
```
"""Server for multithreaded (asynchronous) chat application."""
"""ChatServer.py"""
from socket import AF_INET, socket, SOCK_STREAM
from threading import Thread


def accept_incoming_connections():
   """Sets up handling for incoming clients."""
   while True:
      client, client_address = SERVER.accept()
      print("%s:%s has connected." % client_address)
      client.send(bytes("Greetings from the cave! Now type your name and press enter!", "utf8"))
      addresses[client] = client_address
      Thread(target=handle_client, args=(client,)).start()


def handle_client(client):  # Takes client socket as argument.
   """Handles a single client connection."""

   name = client.recv(BUFSIZ).decode("utf8")
   welcome = 'Welcome %s! If you ever want to quit, type {quit} to exit.' % name
   client.send(bytes(welcome, "utf8"))
   msg = "%s has joined the chat!" % name
   broadcast(bytes(msg, "utf8"))
   clients[client] = name

   while True:
      msg = client.recv(BUFSIZ)
      if msg != bytes("{quit}", "utf8"):
         broadcast(msg, name + ": ")
      else:
         client.send(bytes("{quit}", "utf8"))
         client.close()
         del clients[client]
         broadcast(bytes("%s has left the chat." % name, "utf8"))
         break


def broadcast(msg, prefix=""):  # prefix is for name identification.
   """Broadcasts a message to all the clients."""

   for sock in clients:
      sock.send(bytes(prefix, "utf8") + msg)


clients = {}
addresses = {}

HOST = ''
PORT = 33000
BUFSIZ = 1024
ADDR = (HOST, PORT)
```

```
SERVER = socket(AF_INET, SOCK_STREAM)
SERVER.bind(ADDR)

if __name__ == "__main__":
   SERVER.listen(5)
   print("Waiting for connection...")
   ACCEPT_THREAD = Thread(target=accept_incoming_connections)
   ACCEPT_THREAD.start()
   ACCEPT_THREAD.join()
SERVER.close()
```

"""ChatClient.py"""

```
"""Script for Tkinter GUI chat client."""
from socket import AF_INET, socket, SOCK_STREAM
from threading import Thread
import tkinter


def receive():
   """Handles receiving of messages."""
   while True:
      try:
         msg = client_socket.recv(BUFSIZ).decode("utf8")
         msg_list.insert(tkinter.END, msg)
      except OSError:  # Possibly client has left the chat.
         break


def send(event=None):  # event is passed by binders.
   """Handles sending of messages."""
   msg = my_msg.get()
   my_msg.set("")  # Clears input field.
   client_socket.send(bytes(msg, "utf8"))
   if msg == "{quit}":
      client_socket.close()
      top.quit()


def on_closing(event=None):
   """This function is to be called when the window is closed."""
   my_msg.set("{quit}")
   send()

top = tkinter.Tk()
top.title("Chatter")

messages_frame = tkinter.Frame(top)
my_msg = tkinter.StringVar()  # For the messages to be sent.
my_msg.set("Type your messages here.")
scrollbar = tkinter.Scrollbar(messages_frame)  # To navigate through past messages.
# Following will contain the messages.
```

```
msg_list = tkinter.Listbox(messages_frame, height=15, width=50, yscrollcommand=scrollbar.set)
scrollbar.pack(side=tkinter.RIGHT, fill=tkinter.Y)
msg_list.pack(side=tkinter.LEFT, fill=tkinter.BOTH)
msg_list.pack()
messages_frame.pack()

entry_field = tkinter.Entry(top, textvariable=my_msg)
entry_field.bind("<Return>", send)
entry_field.pack()
send_button = tkinter.Button(top, text="Send", command=send)
send_button.pack()

top.protocol("WM_DELETE_WINDOW", on_closing)

#----Now comes the sockets part----
HOST = input('Enter host: ')
PORT = input('Enter port: ')
if not PORT:
    PORT = 33000
else:
    PORT = int(PORT)

BUFSIZ = 1024
ADDR = (HOST, PORT)

client_socket = socket(AF_INET, SOCK_STREAM)
client_socket.connect(ADDR)

receive_thread = Thread(target=receive)
receive_thread.start()
tkinter.mainloop() # Starts GUI execution.

"""output
>>>python ChatServer.py
Waiting for connection...
127.0.0.1:50190 has connected. """
```



```
Greetings from the cave! Now type your name and press
Welcome user123! If you ever want to quit, type {quit} to
user123: Hello
```

"""Socket Programming and file server.
Program to demonstrate client-server file transfer. """
"""FileServer.py"""

```
import socket                 # Import socket module

port = 60000                  # Reserve a port for your service.
s = socket.socket()           # Create a socket object
host = socket.gethostname()   # Get local machine name
s.bind((host, port))          # Bind to the port
s.listen(5)                   # Now wait for client connection.

print('Server listening....')

while True:
   conn, addr = s.accept()    # Establish connection with client.
   print('Got connection from', addr)
   data = conn.recv(1024)
   print('Server received', repr(data))

   filename='temp.txt'
   f = open(filename,'rb')
   l = f.read(1024)
   while (l):
     conn.send(l)
     print('Sent ',repr(l))
     l = f.read(1024)
   f.close()

   print('Done sending')
   conn.send('connection end')
   conn.close()
```

"""FileClient.py"""

```
import socket                 # Import socket module

s = socket.socket()           # Create a socket object
host = socket.gethostname()   # Get local machine name
port = 60000                  # Reserve a port for your service.
s.connect((host, port))
s.send("Hello server!")

with open('received_file', 'wb') as f:
   print('file opened')
   while True:
      print('receiving data...')
      data = s.recv(1024)
      print('data=%s', (data))
      if not data:
         break
      # write data to a file
      f.write(data)
```

```
f.close()
print('Successfully get the file')
s.close()
print('connection closed')
```

"""Program to download a webpage. """

```
from urllib.request import urlopen
html = urlopen("http://www.google.com/")
print(html.read())
"""output
Code of entire web page
"""
```