

Practical No: 1

Topic: File Handling.

1 Write a program to copy one file into another.

→ Program:

```
ip = open("data.txt", "r")
op = open("copy.txt", "w")
for line in ip:
    op.write(line)
ip.close()
op.close()
ip = open("copy.txt", "r")
line = ip.read()
print(line)
```

Output:

```
copy.txt
Hello World
```

2. Write a program to count the number of lines and words in a file.

→ Program:

```
ip = open("myfile.txt", "r")
nol = 0
```

```
now = 0
nwc = 0
line = ip.readline()
while line != "":
    nol = nol + 1
    lst = line.split(" ")
    now = now + len(lst)
    line = ip.readline()
print ("No of lines = ", nol)
print ("No of words = ", now)
```

Output:

myfile.txt
Hello World

No of lines = 1

No of words = 2

3. Write a program in python to count the number of spaces in a file.



Program:

```
ip = open("data.txt", "r")
line = ip.readline()
num_of_spaces = 0
while line != "":
```

```
for chr in line:  
    if chr == ' ':  
        num_of_spaces = num_of_spaces + 1  
line = ip.readline()  
print("Number of spaces:", num_of_spaces)
```

Output:

data.txt

Python is a programming language

Number of spaces: 4

4. Write a program to add the contents of one file into another.

→ Program:

```
op=open("data1.txt", "a")  
ip=open("data2.txt", "r")  
line = ip.readline()  
while line != '':  
    op.write(line)  
    line = ip.readline()  
ip.close()  
op.close()  
ip=open('data1.txt', "r")  
line = ip.read()
```

```
print (line)
```

Output:

data1.txt

Hello World

data2.txt

Enjoy programming.

data1.txt

Hello World

Enjoy programming.

-
5. Write a program to convert the contents of file to uppercase.



Program:

```
ip = open ("data.txt")
op = open ("tmp.txt", "w")
line = ip.readline()
while (line != ''):
    capline = line.upper()
    op.write (capline)
    line = ip.readline()
ip.close()
op.close()
import os
os.remove ("data.txt")
```

```
os.rename ("tmp.txt", "data.txt")
```

Output:

data.txt

hello world

tmp.txt

HELLO WORLD

6. Write a program that reads a text file and write every other line starting from the first line, to a new file name half_text.

→ Program:

```
ip=open ('data.txt')
op=open ('half_text.txt', 'w')
cnt = 1
line = ip.readline()
while line != '':
    if cnt == 1:
        op.write (line)
        cnt=0
    elif cnt == 0:
        cnt = 1
    line = ip.readline()
ip.close()
op.close()
```

Output:

data.txt
Hello World
half-test.txt
Hello World

2. Write a program to sort (arrange in ascending order) the numbers stored in a file.

→ Program:

```
fp = open("num.txt", "r+")
line = fp.readline()
lst = []
while line != "":
    lst.append(int(line))
    line = fp.readline()
lst.sort()
fp.seek(0)
for k in lst:
    fp.write(str(k) + "\n")
fp.close()
```

Output:

num.txt

5

1

Expt. No. :



Page No. : 4

Date : 1 1

2

4

3

After execution:

num.txt

1

2

3

4

5

✓

Practical No: 2.

Topic : Exceptions

1. Execute the following in Python shell and write output.

i) `>>> lst = [1, 2, 3]`

`>>> lst[3]`

→ Index Error

ii) `>>> 2 + '3'`

→ Type Error

iii) `>>> lsst[0]`

→ Name Error

iv) `>>> int('12.04')`

→ Datatype Error

v) `>>> import rem`

→ Import Error

vi) `>>> open('xyz')`

→ FileNotFoundError

vii) `>>> fp = open('data.txt')`

`>>> fp.write('hello')`

→ IO Error

2. Write a program to compute the factorial of a number entered through the keyboard. If the number entered is negative, handle the appropriate exception and accept

the input again.

→ Program:

```
import math
num = int(input('Enter a number to compute a factorial of:'))
valid_input = False
while not valid_input:
    try:
        result = math.factorial(num)
        print('Factorial of ', num, ' is ', result)
        valid_input = True
    except ValueError:
        print('Cannot compute factorial of negative number')
num = int(input('Re-enter:'))
```

Output:

Enter a number to compute a factorial of: 5
Factorial of 5 is 120

Enter a number to compute a factorial of: -5
Cannot compute factorial of negative number
Re-enter: 4

Factorial of 4 is 24

3. Write a program to display the contents of a file. The

file name is entered through the keyboard. If the file does not exist, handle the appropriate exception and take another file name from user.

→ Program:

```
file_name = input('Enter the file name :')
file_opened = False
while not file_opened:
    try:
        ip = open(file_name, 'r')
        file_opened = True
        line = ip.readline()
        while line != '':
            print(line)
            line = ip.readline()
    except FileNotFoundError:
        print('File does not exist')
        file_name = input('Enter another file name :')
```

Output:

Enter the file name : data.txt

Hello World.

Enter the file name : xyz.txt

FileNotFoundException: File does not exist

4. Write a program to take a month number from the user and raise an appropriate exception if the month

is not valid range (1-12), otherwise display the month name corresponding to the month number.

→ Program:

```
m = int(input('Enter a month number (1-12) : '))
valid_month = False
while not valid_month:
    try:
        if m < 1 or m > 12:
            raise ValueError('Not a valid month')
    except ValueError as e:
        print("Error!", e)
        m = int(input('Enter month again: '))
    else:
        valid_month = True
    if m == 1:
        month_name = 'January'
    elif m == 2:
        month_name = 'February'
    elif m == 3:
        month_name = 'March'
    elif m == 4:
        month_name = 'April'
    elif m == 5:
        month_name = 'May'
    elif m == 6:
        month_name = 'June'
```

```
elif m == 7:
```

```
    month_name = 'July'
```

```
elif m == 8:
```

```
    month_name = 'August'
```

```
elif m == 9:
```

```
    month_name = 'September'
```

```
elif m == 10:
```

```
    month_name = 'October'
```

```
elif m == 11:
```

```
    month_name = 'November'
```

```
elif m == 12:
```

```
    month_name = 'December'
```

```
print('Month name : ' + month_name)
```

Output:

```
Enter a month number : 10
```

```
Month name: October
```

```
Enter a month number : 15
```

```
ValueError: Not a valid month
```

```
Error!
```

```
Enter month again: 7
```

```
July.
```

Write a program to find prime nos between m and n, entered by the user. If $m > n$, handle appropriate



exception and accept m and n again.

Program:

```
m = int(input('Enter m:'))
n = int(input('Enter n:'))
valid_input = False
while not valid_input:
    try:
        if m > n:
            raise ValueError('m should be less than n')
        valid_input = True
    i = m
    while i <= n:
        j = 2
        while j < i:
            if i % j == 0:
                break;
            j += 1
        if j == i:
            print(i)
        i += 1
    except ValueError as e:
        print("Error!", e)
        print('Re-enter:')
        m = int(input('Enter m:'))
        n = int(input('Enter n:'))
```



Page No. : 8

Date : 1 1

Expt. No. :

Output:

~~0~~ Enter m: 1

Enter n: 9

2

3

5

7

✓

Practical No : 3

Topic: Regular Expressions.

1. Write a program to accept an email address from the user and determine if the email address is correctly formed. If the email address is correct, display the username and the domain name from the email address.

→ Program:

```
import re
email = input('Enter email address : ')
pattern = '([\w.-]+)@([\w.-]+)\$'
m = re.match(pattern, email)
if m:
    print('You have entered correct email address')
    print('Username : ', m.group(1))
    print('Domain name : ', m.group(2))
else:
    print('It is incorrect')
```

Output:

```
Enter email address : xyz16@gmail.com
You have entered correct email address
User name : xyz16
Domain name : gmail.com
```

2. Write a program to convert date yyyy-mm-dd format to dd-mm-yyyy format.

→ Program:

```
import re  
dt = '2016-12-16'  
newdt = re.sub('(\d{4})-(\d{2})-(\d{2})',  
              r'\3-\12-\2', dt)  
print(newdt)
```

Output:

16-12-2016

3. Consider a file consisting of three column : name, email-id and mobile no. Write a program to read the file and store email-ids in one file and mobile numbers in another file.

→ Program:

```
import re  
pattern1 = '\d{10}'  
pattern2 = '([\w.-]+)@([\w.-]+)'  
ip = open('data.txt')  
op1 = open('phone.txt', 'w')  
op2 = open('email.txt', 'w')  
line = ip.readline()  
while line != '':
```

```
m = re.search(pattern1, line)
if m:
    op1.write(m.group(0) + '\n')
m = re.search(pattern2, line)
if m:
    op2.write(m.group(0) + '\n')
line = ip.readline()
ip.close()
op1.close()
op2.close()
```

Output:

data.txt

4. Write a program to convert date entered in the form dd/mm/yy into word form (dd month-name yyyy)

→ Program:

```
import re
dt = '09/12/16'
```

```
d = re.match('(\d{2})/', dt)
m = re.search('/(\d{2})/', dt)
y = re.search('/(\d{2},4?)$', dt)
mm = int(m.group(1))
if mm == 1:
    mon = 'January'
elif mm == 2:
    mon = 'February'
elif mm == 3:
    mon = 'March'
elif mm == 4:
    mon = 'April'
elif mm == 5:
    mon = 'May'
elif mm == 6:
    mon = 'June'
elif mm == 7:
    mon = 'July'
elif mm == 8:
    mon = 'August'
elif mm == 9:
    mon = 'September'
elif mm == 10:
    mon = 'October'
elif mm == 11:
    mon = 'November'
```

```
elif mm==12:  
    mon='December'  
yy=y.group(1)  
if len(yy)==2:  
    yy='20'+yy  
print(d.group(1),mon,',',yy)
```

Output:

09 December 2016

5. Write a program to read a line of text from the user and print the number of vowels and number of consonants in the line.

→ Program:

```
line=input('Enter a line of text: ')  
pattern1='[aeiou]'  
pattern2='[^aeiou\s]'  
import re  
lst=re.findall(pattern1,line,re.IGNORECASE)  
print('No of vowels are:',len(lst))  
lst=re.findall(pattern2,line,re.I)  
print('No of consonants are:',len(lst))
```

Output:

Enter a line of text: welcome to python

No of vowels are : 5
No of consonants are : 10

6. Write a program to remove multiple occurrences of spaces from a file using the concept of regular expressions.



Program:

```
import re
ip = open('data.txt')
op = open('tmp.txt', 'w')
line = ip.readline()
while line != '':
    newline = re.sub(' +', ' ', line)
    op.write(newline)
    line = ip.readline()
ip.close()
op.close()
```

Output:

data.txt
Welcome to Python

tmp.txt
Welcome to Python

7. Write a program to accept a number from the user and determine if it is correct decimal number of precision two.

→ Program:

```
import re
num = input('Enter a decimal number: ')
pattern = '\d+\.\d{1,2}$'
m = re.match(pattern, num)
if m:
    print('Decimal number is of precision two')
else:
    print('Its incorrect')
```

Output:

Enter a decimal number: 2.55

Decimal number is of precision two

Enter a decimal number: 2.621

Its incorrect.

8. Write a program to accept a string and determine if it is correctly formed variable name.

→ Program:

```
import re
num = input('Enter a variable name: ')
pattern = '[a-zA-Z][a-zA-Z0-9_]*$'
```

```
m=re.match(pattern,num)
if m:
    print('Correct variable name')
else:
    print('Its incorrect')
```

Output:

Enter a variable name: month-name
Correct variable name
Enter a variable name: month.name/
Its incorrect.

2

Practical No: 4**Topic: Canvas Widget**

1. Write a program to draw the following:

i. cone

→ Program:

```
from tkinter import *
```

```
win = Tk()
```

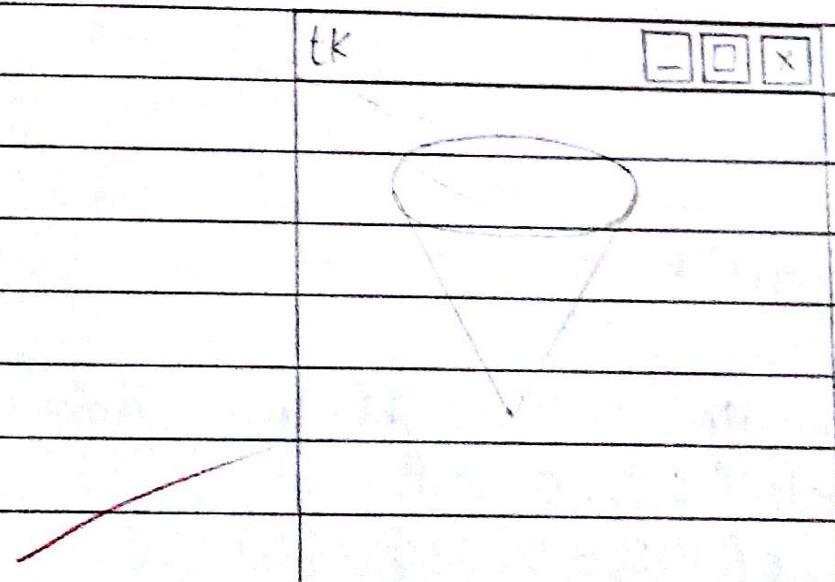
```
c = Canvas (win, width = 400, height = 400, bd = 10)
```

```
c.create_oval (100, 50, 300, 100)
```

```
c.create_line (100, 80, 190, 280)
```

```
c.create_line (300, 80, 190, 280)
```

```
c.pack()
```



ii. cylinder

→ Program:

```
from tkinter import *
```

```
win = Tk()
```

```
c = Canvas(win, width=400, height=400, bd=10)
```

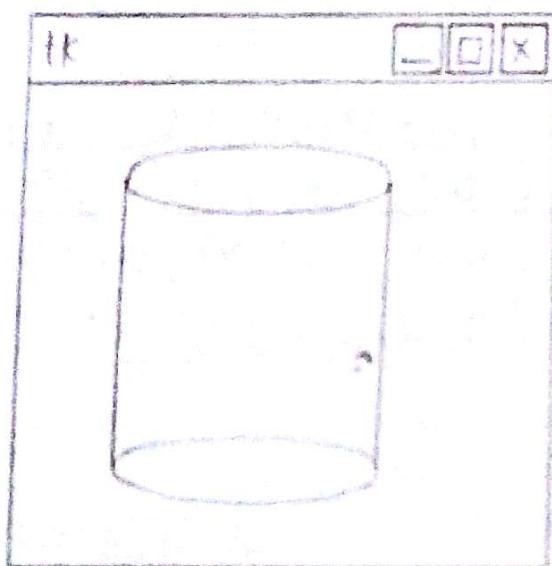
```
c.create_oval(77, 53, 197, 96)
```

```
c.create_oval(187, 179, 197, 220)
```

```
c.create_line(76, 74, 76, 199)
```

```
c.create_line(197, 75, 197, 199)
```

```
c.pack()
```



iii. cube

→ Program:

```
from tkinter import *
```

```
win = Tk()
```

```
c = Canvas(win, width=400, height=400, bd=10)
```

```
c.create_rectangle(60, 20, 170, 130)
```

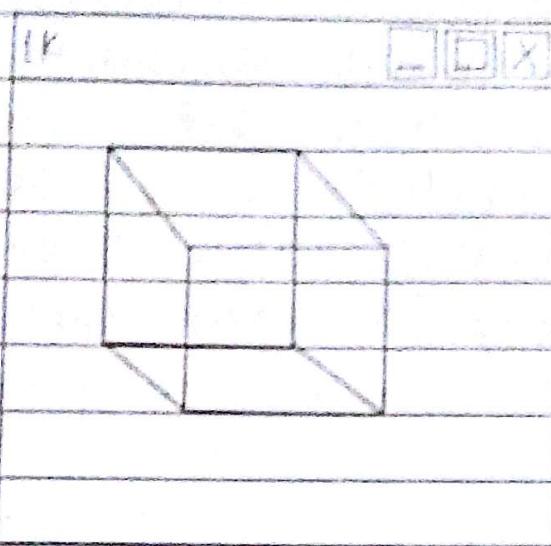
```
c.create_rectangle(100, 50, 210, 160)
```

```
c.create_line(60, 20, 100, 50)
```

```
c.create_line(170, 20, 210, 50)
```

```
c.create_line(60, 130, 100, 160)
```

c.create_line(170, 130, 210, 160)
c.pack()

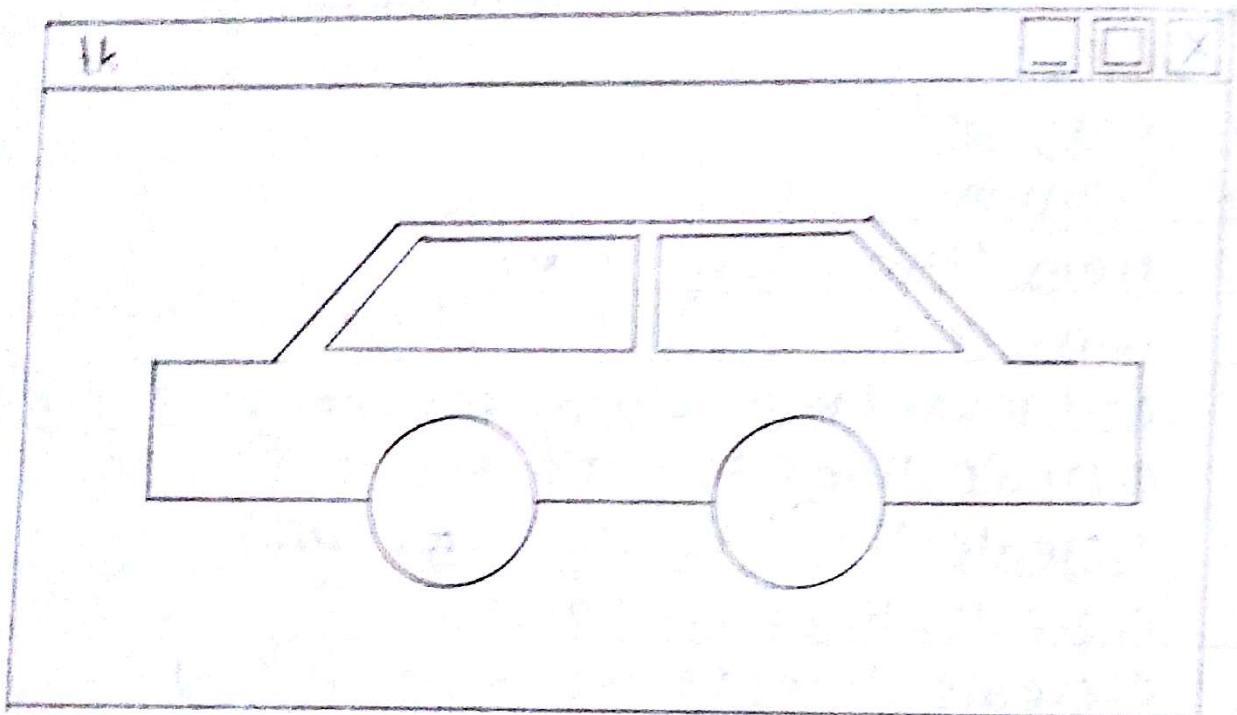


IV. CAR

→ Program:

```
from tkinter import *
win = Tk()
c = Canvas(win, width=800, height=800, bd=10)
c.create_line(300, 100, 500, 100)
c.create_line(300, 100, 250, 250)
c.create_line(50, 150, 150, 150)
c.create_line(50, 150, 50, 250)
c.create_line(50, 250, 750, 250)
c.create_line(750, 150, 750, 250)
c.create_line(650, 150, 750, 150)
c.create_line(500, 100, 650, 150)
c.create_line(300, 120, 400, 120)
c.create_line(400, 120, 400, 150)
```

```
c.create_line(200,150,400,150)
c.create_line(300,120,200,150)
c.create_line(420,120,500,120)
c.create_line(500,120,600,150)
c.create_line(420,150,600,150)
c.create_line(420,120,420,150)
c.create_oval(120,220,210,290)
c.create_oval(520,220,620,290)
c.pack()
```



2.

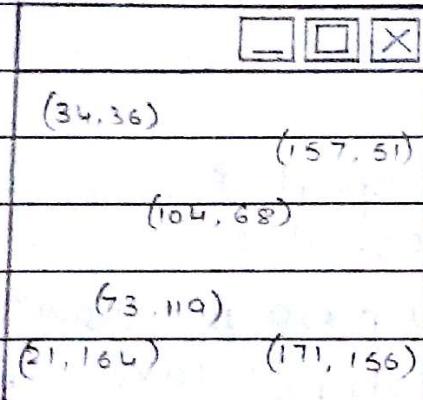
i. Write a program to display the co-ordinates where mouse is clicked

→ Program:

```

from tkinter import *
win = Tk()
c = Canvas(win, width=200, height=200, bg='lightblue')
c.bind("<Button-1>", display)
c.pack()
def display(event):
    s = ('(' + str(event.x) + ', ' + str(event.y) + ')')
    c.create_text(event.x, event.y, text=s)

```



ii. Write a program to display a dot where the mouse is clicked.

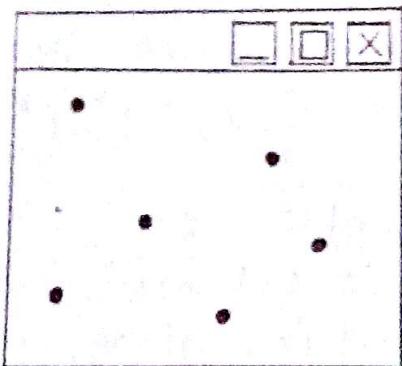
→ Program:

```

from tkinter import *
def display(event):
    x = event.x
    y = event.y
    c.create_oval(x-5, y-5, x+5, y+5, fill='black')
win = Tk()
c = Canvas(win, width=200, height=200, bg='lightblue')
c.bind("<Button-1>", display)

```

c.pack()

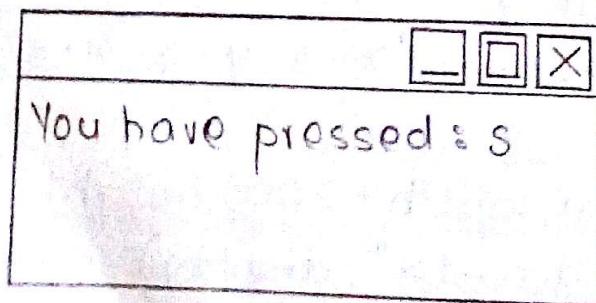


iii. Write a program to display which key is pressed in canvas.



Program:

```
from tkinter import *
def display(event):
    str = "You have pressed:" + event.char
    c.itemconfig(t, text=str)
win = Tk()
c = Canvas(win, width=200, height=200, bg='light blue')
t = c.create_text(100, 100, text='press any key')
c.bind("<key>", display)
c.focus_set()
c.pack()
```



3. Write a program that changes the colour of rectangle to red when the mouse is moved inside the rectangle and changes it to blue when the mouse is moved outside the rectangle.

Program:

```
from tkinter import *
def paint(event):
    if event.x >= x1 and event.y >= y1 and
       event.x <= x2 and event.y <= y2 :
        c.itemconfig(r, fill = 'blue')
    else:
        c.itemconfig(r, fill = 'red')
```

win = Tk()

c = Canvas(win, height = 400, width = 400)

x1 = 100

y1 = 100

x2 = 300

y2 = 400

r = c.create_rectangle(x1, y1, x2, y2)

c.bind('<Motion>', paint)

c.pack()

4. Write a program to increase the font of the text displayed in a canvas when key 'i' is pressed and decrease the font when key 'control-i' is pressed from keyboard.

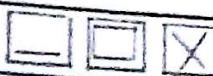
→ Program:

```
from tkinter import *
def incrsize(event):
    i = t.cget("size")
    if i < 40:
        i += 2
    t.config(size=i)
    e.itemconfig(t, font=f)
def decrsize(event):
    i = t.cget("size")
    if i >= 10:
        i -= 2
    t.config(size=i)
    e.itemconfig(t, font=f)
```

win = Tk()

```
c = Canvas(win, width=500, height=500)
f = font.Font(family='Times', size=12)
t = c.create_text(100, 100, text='Hello', font=f)
c.bind("<key-i>", incrsize)
c.bind("<key-l>", incrsize)
c.bind("<Control-key-i>", decrsize)
c.bind("<Control-key-l>", decrsize)
c.focus_set()
c.pack()
```

TK



TK



Hello World

Hello World

5. Write a program to change the color of canvas when mouse enters or leaves the canvas.



Program:

```
from tkinter import *
def paint1(event):
    c.config(bg='red')
def paint2(event):
    c.config(bg='blue')
win = Tk()
c = Canvas(win, height=400, width=400)
c.bind('<Enter>', paint1)
c.bind('<Leave>', paint2)
c.pack()
```

B
14/3/17