UNIT IV

FILES, MODULES, PACKAGES

Files: text files, reading and writing files, format operator; modules, packages; Illustrative programs: word count, copy.

File I/O

1. **Write a Python program to perform read and write operations on a file.**

1.1

```
str =input("Enter the data into a file : ")
f1=open('abc.txt','w')
f1.write(str)
f1.close()
f2=open('abc.txt','r')
data=f2.read()
print(data)
f2.close()
```

```
Enter the data into a file : Python Programming Lab Manual
Python Programming Lab Manual
```

1.2

```
filename =input("Enter the file name : ")
f1=open(filename,'w')
f1.write('Computer science Engineering\n')
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f2=open(filename,'r')
data=f2.read()
print(data)
f2.close()
```

```
Enter the file name : abc
Computer science Engineering
Elecronics and Communication Engineering
Civil Engineering
```

1.3

```
filename =input("Enter the file name : ")
f1=open(filename,'w')
f1.write('Computer science Engineering\n')
```

```
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f2=open(filename,'r')
data=f2.readline()
print(data)
f2.close()
```

```
Enter the file name : rgm
Computer science Engineering
```

1.4

```
filename =input("Enter the file name : ")
f1=open(filename,'w')
f1.write('Computer science Engineering\n')
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f2=open(filename,'r')
data=f2.readlines()
print(data)
f2.close()
```

```
Enter the file name : Python
['Computer science Engineering\n', 'Elecronics and Communication Engineeri
ng\n', 'Civil Engineering\n']
```

2. **Write a Python program to copy the contents of one file to another file.**

```
filename1 = input("Enter the file name 1: ")
f1 = open(filename1,'w')
f1.write('Computer science Engineering\n')
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f2 = open(filename,'r')
data1 = f2.read()
print(data)
f2.close()
filename2 = input("Enter the file name 2: ")
f3 = open(filename2,'w')
f3.write(data1)
f3.close()
f3 = open(filename2,'r')
data2 = f3.read()
print(data2)
f3.close()
```

```
Enter the file name 1: Pyhton
['Computer science Engineering\n', 'Elecronics and Communication Engineeri
ng\n', 'Civil Engineering\n']
Enter the file name 2: c++
Computer science Engineering
Elecronics and Communication Engineering
Civil Engineering
```

3. **Write a Python program to count frequency of characters in a given file.**

```
filename = input("Enter a file name : ")
f1 = open(filename,'w')
f1.write('Computer science Engineering\n')
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f1 = open(filename,'r')
data = f1.read()
a =list(set(data))
sorted(a)
print(a)
f1.close()
for i in a:
      print("{} as occured {} times".format(i,data.count(i)))
```

```
Enter a file name : xyz
[' ', 'r', 'E', 'C', 't', 'p', 'g', 'u', 'o', 'c', 'm', 'a', 'i', '\n',
's', 'v', 'e', 'l', 'n', 'd']
  as occured 6 times
r as occured 5 times
E as occured 4 times
C as occured 3 times
t as occured 2 times
p as occured 1 times
g as occured 6 times
u as occured 2 times
o as occured 4 times
c as occured 5 times
m as occured 3 times
a as occured 2 times
i as occured 12 times
 as occured 3 times
s as occured 2 times
```

```
v as occured 1 times
e as occured 10 times
l as occured 2 times
n as occured 14 times
d as occured 1 times
```

4. **Write a Python program to print each line of a file in reverse order.**

```
filename = input("Enter a file name : ")
f1 = open(filename,'w')
f1.write('Computer science Engineering\n')
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f1 = open(filename,'r')
data = f1.readlines()
print(data)
for i in data:
   print(i[::-1])
f1.close()
```

```
Enter a file name : bcd
['Computer science Engineering\n', 'Elecronics and Communication Engineeri
ng\n', 'Civil Engineering\n']
gnireenignE ecneics retupmoC
gnireenignE noitacinummoC dna scinorcelE
gnireenignE liviC
```

5. **Write a Python program to compute the number of characters, words and lines in a file.**

```
filename = input("Enter a file name : ")
c=0
w=0
l=0
f1 = open(filename,'w')
f1.write('Computer science Engineering\n')
f1.write('Elecronics and Communication Engineering\n')
f1.write('Civil Engineering\n')
f1.close()
f2 = open(filename,'r')
data = f2.readlines()
print(data)
for i in data:
      c = c + len(i)
      w = w + len(i.split())
      l=l+1
f2.close()
```

```
print('The number of Characters are :', c)
print('The number of Words are :', w)
print('The number of Lines are :', l)
```

```
Enter a file name : cde
['Computer science Engineering\n', 'Elecronics and Communication Engineeri
ng\n', 'Civil Engineering\n']
The number of Characters are : 88
The number of Words are : 9
The number of Lines are : 3
```

6. **Write a Python program to reverse first n characters in a file.**

```
f = open("abc",'r+')
n = int(input("Enter the first n number to reverse in a file : "))
s =f.read(n)
print(s)
f.seek(0,0)
f.write(s[n::-1])
f.close()
f = open("abc",'r')
f.seek(0,0)
print(f.read())
f.close()
```

```
Enter the first n number to reverse in a file : 5
upmoC
Computer science Engineering
Elecronics and Communication Engineering
Civil Engineering
```

MODULES:

- Modules are pre-defined set of code that is used for performing certain common tasks like random number generation, mathematical operation, etc.

- A Python module is a file that contains one or more related functions or executable code that can be included in any applications.
- Modules can be reused easily or shared with other programmers.

CREATING A MODULE:

- To create a module, write the code / functions in a file and save it in .py extension. To illustrate, this code is saved as greet_mod.py.

```
def greeting (name):
        print ("Hi!" + name + "! This is a sample module")
```

IMPORTING THE MODULE:

- To use a module, just import the .py module using **import** or **from** keyword and call the function inside the module. For the above example, the code is written using **import** keyword as follows:

```
import greet_mod
greet_mod.greeting("JITians")
```

- The above code is saved as call_mod.py. This is run by pressing F5. The output is as follows:

```
>>>
 RESTART: A:/JIT/2018 - 19/Odd Semester/PY Lab - I - LF/Python Programs/call_mod
.py
Hi!JITians! This is a sample module
```

- The code can also be written using **from** keyword. The syntax for using 'from' keyword is:

from <module_name> import <function or variable or *>

- The example can be rewritten as:

```
from greet_mod import *
greeting("JITians")
```

VARIABLES IN MODULES:

- The modules can contain not only functions but also variables of all types. The code in greet.py can be modified to include variables as:

```
def greeting (name):
    print ("Hi!" + name + "! This is a sample module")
height = 156
weight = 56
```

- While accessing the variables, import command is used in call_mod.py and the variables are accessed as follows:

```
import greet_mod
greet_mod.greeting("JITians")
ht = greet_mod.height
wt = greet_mod.weight
print ("Height is :",ht)
print ("Weight is: ",wt)
```

- The output of the code will be as follows:

```
>>>
 RESTART: A:\JIT\2018 - 19\Odd Semester\PY Lab - I - LF\Python Programs\call_mod
.py
Hi!JITians! This is a sample module
Height is : 156
Weight is:  56
```

- In the above example, the variables are used. We can also use tuples, lists or dictionaries.

RENAMING THE MODULE:

- An alias name can be created for the module while importing the module, by using the 'as' keyword.
- The call_mod.py is modified as:

```
import greet_mod as gm
gm.greeting("JITians")
ht = gm.height
wt = gm.weight
print ("Height is :",ht)
print ("Weight is: ",wt)
```

- This code will produce the same output

BUILT-IN MODULES:

- There are many built – in modules in Python that can be imported whenever there is a need.
- To use a built – in Python module, just import that module and start using it.
- Some of the most used built – in modules are:
    - Math – it consists of various functions to perform mathematical operations
    - Sys – this module is used to access system variables and functions
    - Calendar – consists of functions to work with calendars
    - Random – used for generating random numbers
    - Sysconfig – used for accessing Python's configuration information
- Example 1 – using Math module

```
#Program to find the area of circle using in-built Module
import math
radius = int (input ("Enter the radius of the Circle: "))
aoc = math.pi * radius * radius
print("The area of the Circle is: ", aoc)
```

```
Enter the radius of the Circle: 5
The area of the Circle is:  78.53981633974483
```

- Example 2 – using Calendar module

```
#Program to display calendar using in-built Module
import calendar
yy = int(input("Enter the year: "))
mm = int(input("Enter month: "))
print(calendar.month(yy, mm))
```

```
Enter the year: 2014
Enter month: 5
      May 2014
Mo Tu We Th Fr Sa Su
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```