

## ▼ Experiment No : 4

Dated : 4th Feb 2022

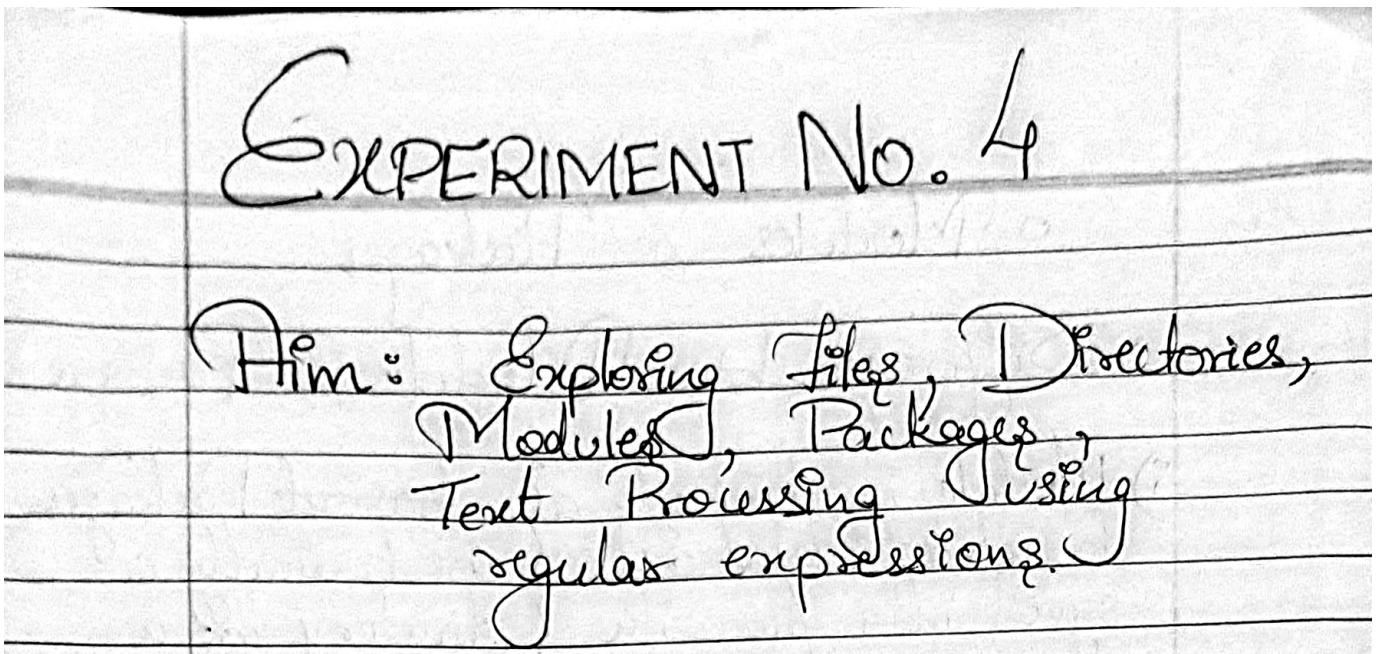
## ▼ Aim

Exploring Files, Directories, Modules, Packages, Text Processing using regular expressions

## ▼ Theory

- Files & Directories
  - Enlist the basic File operations
  - Enlist the basic operations on Directories (at least 4)
- Modules & Packages
  - Difference between Modules & Packages (at least 2)
- Text Processing & Regular Expressions
  - Enlist at least 4 functions in re module

## ▼ Handwritten Theory



## Theory: Files & ~~Dir~~ Directories

### 1) Basic File operations

- a) Open a file
- b) Read or write (perform operation)
- c) Close the file.

### 2) Basic operations on directories.

- a) Creating a directory
- b) Renaming a directory
- c) Listing all directories
- d) Removing a directory

## ◦ Modules & Packages

### \* Difference btw Module & Package

- i) Module is basically a bunch of related code saved in a file with the .py extension
  - ii) Used to define functions, classes, or variables in a module
- are basically a directory of a collection of modules
- allows hierarchal structure of the module namespace.

## ◦ Text Processing & Regular Expressions

### \* 4 functions in re module

- i) compile,
- ii) search,
- iii) match,
- iv) span,
- v) end,
- vi) start.

## ▼ Programs to be performed :

### ▼ Exploring File Handling (attempt any 2)

1. Python program to append data to existing file and then display the entire file
2. Python program to count the word frequency in a file.
3. Python program to open a file & convert the text in a file to upper case and save it.
4. Python program to find the most repeated word in a text file

```
file1 = open("file1.txt", 'w')#create a file using write mode
L = ["This is line 1.\n","This is line 2.\n","This is line 3.\n"]
file1.writelines(L)
file1.close()

file1 = open("file1.txt", "r") #displaying the contents of the file before appending
print(file1.read())
file1.close()

file1 = open("file1.txt","a") #opening the file in append mode
file1.write("This is line 4.\n") #appended line
file1.close()

file1 = open("file1.txt", "r") #displaying the contents of the file after appending
print(file1.read())
file1.close()
```

```
This is line 1.
This is line 2.
This is line 3.
```

```
This is line 1.
This is line 2.
This is line 3.
This is line 4.
```

```
file2 = open("file2.txt", "w")
L = ['Mango'," ", 'banana'," ", 'apple'," ", 'pear'," ",
     'Banana'," ", 'grapes'," ", 'strawberry'," ",
     'Apple'," ", 'pear'," ", 'mango'," ", 'banana'," ",
     'Kiwi'," ", 'apple'," ", 'mango'," ", 'strawberry']
file2.writelines(L)
file2.close()

file2 = open("file2.txt", "r")
print(file2.read())
file2.close()
```

```
# Create an empty dictionary
d = dict()

file2 = open("file2.txt", "r")
# Loop through each line of the file
for line in file2:
    # Remove the leading spaces and newline character
    line = line.strip()

    # Convert the characters in line to
    # lowercase to avoid case mismatch
    line = line.lower()

    # Split the line into words
    words = line.split(" ")

    # Iterate over each word in line
    for word in words:
        # Check if the word is already in dictionary
        if word in d:
            # Increment count of word by 1
            d[word] = d[word] + 1
        else:
            # Add the word to dictionary with count 1
            d[word] = 1
file2.close()
# Print the contents of dictionary
for key in list(d.keys()):
    print(key, ":", d[key])
```

```
Mango banana apple pear Banana grapes strawberry Apple pear mango banana Kiwi app
mango : 3
banana : 3
apple : 3
pear : 2
grapes : 1
strawberry : 2
kiwi : 1
```

## ▼ Exploring Directories (attempt any 2)

1. Python program to display file available in current directory
2. Write a program to list all files in the given directory
3. Write a program extcount.py to count the number of files for each extension in the given directory. The program should take a directory name as argument and print count and extension for each available file extension.
4. Write a program to print directory-tree. The program should take the path of a directory as an argument and print all the files in it recursively as a tree.

```

import random
import datetime

#start: it is the star number in a range. i.e., lower limit. The default value is 0 if
#stop: It is the end/last number in a range. It is the upper limit.

print("Random number between 0 and 6 (excluding 6) : ", end = "")
print(random.randrange(start = 0, stop = 6))

print("Random number between 5 and 10 (excluding 10) : ", end = "")
print(random.randrange(start = 5, stop = 10))

#step : Specify the increment value in range. The generated random number divisible by

print("Random numbers between 0 and 10 with a step of 3 : ", end = "")
print(random.randrange(start = 0, stop = 10, step = 3))

print("Random date between two dates : ", end = "")

starting_date = datetime.date(2022, 2, 13)
ending_date = datetime.date(2022, 3, 13)
tdelta = ending_date - starting_date
tdelta_days = tdelta.days
random_dayss = random.randrange(tdelta_days)
random_dates = starting_date + datetime.timedelta(days = random_dayss)
print(random_dates)

```

```

Random number between 0 and 6 (excluding 6) : 4
Random number between 5 and 10 (excluding 10) : 5
Random numbers between 0 and 10 with a step of 3 : 3
Random date between two dates : 2022-03-01

```

## ▼ Modules & Packages (attempt any 1)

1. Create a package named "arithmetic\_op" and add modules named, "add.py", "diff.py", "square.py" and "modulo.py". Demonstrate the usage of these modules in your program.
2. Write a Python program to generate a random integer between 0 and 6 - excluding 6, random integer between 5 and 10 - excluding 10, random integer between 0 and 10, with a step of 3 and random date between two dates. Use random.randrange()
3. Write a Python program to configure the rounding to round to the floor, ceiling. Use decimal.ROUND\_FLOOR, decimal.ROUND\_CEILING
4. Write a Python program to read and display the content of a given CSV file. Use csv.reader

## ▼ Text Processing (attempt any 1)

1. Python program to print even length words in a string

2. Python program to create a list of words from a given string.
3. Python program that will read a given text through each line and look for sentences. Print each sentence and divide two sentences with "=====".
4. Python program to remove stopwords from a Text.

```
def even_length_of_words(words):  
  
    words = words.split()  
  
    print("Even length of words in the given string are : ", end = "")  
    for word in words:  
        if len(word) % 2 == 0:  
            print(word, end = " , ")  
  
words = input("Enter the sentence : ")  
even_length_of_words(words)
```

```
Enter the sentence : Hehehe Monke  
Even length of words in the given string are : Hehehe ,
```

✓ 0s completed at 12:43 AM

