

## Task 6 Implement various text file operation

Aim: TO write a Python program implement various text file operation.

Program 6.1: You need to write the sentence "Error objects are thrown when running time error occur the error object can also be used as a base object for users-defined exceptions" into a text file named log.txt implement a function that performs this task

Algorithm:

1) write a file:

- Define writefile (filename) function;
- Open a file named "log.txt" in write mode.
- write the following txt to the file:  
"Error objects are thrown when tm times errors occur the error object can also be used as a base object for user defined exception"
- Close the file.

2) Read from a file:

- Define read file (filename) function;
- Open the file specified by file name in read mode using a with statement
- Read the entire content of the file.
- Print the content.

3) Execute the program

- call write file ("write") to write the predefined text to "log.txt".
- call readfile ("text")\* to attempt to read from a file named "text" and print its content

Monteiro (1999) notes a trend of long  
distances with high visibility.

Output- Error objects are thrown when run time errors occur. The error object can also be used as base objects for user-defined exceptions.

• benefits effective after some time  
• shows strikes in "fxn poly" between file & ref ID  
• ref ID is fxn polynomial fitted to the file  
• generates random words from existing words  
• can be used as a perfect hash function  
• possible object to benefit from it  
• 1917 0 1917 92011

all stirs of ("Coxins") stir stirs less  
"fixed" of free bonitoberg  
at base of stems of "Cox" I didn't see.  
stirrers in fair but "fixed" former stirrers

## Program 6.1

```
def write_file (filename):
    f = open ('log.txt', "w")
    f.write ("Error objects are thrown when
    run time errors occurs the error object
    can also be used as a base object
    for user-defined exceptions). f.close()

def read_file (filename):
    with open (filename, "r") as file:
        content = file.read()
        print (content)
        file.write ("write")
        file.read ("text")
```

- ① check each line for error objects
- ② loop through each line in the file
- ③ if the line contains the word error  
increment error count by 1
- ④ return error count
- ⑤ after reading all the lines return the value of error count
- ⑥ execute the program
- ⑦ enter error message line ("log.txt") to count the value of error in the file.
- ⑧ after the execution the message will be displayed in the message window

## 6.2 Program

You have a text file log.txt containing logs of a system. write a function let that counts the number of lines containing the word "ERROR".

### Algorithm:

#### 1) Initialize Error counter:

- Define the function count-error-line (file-name);

#### 2) Open and Read file:

- Open the file specified by file name in read mode using a with statement

#### 3) Check each line for "ERROR":

loop through each line in the file:

- If the line contains the word "ERROR", increment error-count by 1

#### 4) Return Error count:

- After reading all the lines return the value of error-count

#### 5) Execute the program:

- Call count-error-lines ("log.txt") to count number of lines with "error" in the file "log.txt"

- Print the result with the message: "Number of lines with 'ERROR':

{error-lines?}"

Output:-

Number of lines with "Error" is 2

Below error count :-

1) Error - 001

2) Error - 002

```

def count_error_lines(filename):
    error_count = 0
    with open(filename, "r") as file:
        for line in file:
            if "ERROR" in line:
                error_count += 1
    return error_count
error_lines = count_error_lines
print(f"Number of times with ERROR"
      f"\n{error_lines}")

```

`dog.txt`

"`ERROR` objects are thrown when run time error occur  
 the error object can also be used as a base object for user defined exceptions"

Program :- 6.3

You need to write the report containing the details (Name, departments) of the employee in list write a python function that write this report to a file named `employee-report.txt`.

Algorithm :- 1) create employee Data:

- Define the function `write_employee_report(filename)`:
- Create a list `employee` containing dictionaries, each with "name" and "department" keys for individual employee.

- 2) Open file for writing
- open the file specified by file name in write mode using a with statement.
- 3) Write employee data to file.
- loop through each employee in the employees list
  - for each employee, format a string as "Name: \${employee['name']}, Department: \${employee['department']}
  - write the formatted string to the file, followed by a newline character ('n')
- ~~4) Execute the program~~

### Program 6.3

```

def write_employee_report(filename):
    employees = [
        {"name": "Alice", "department": "HR"},
        {"name": "Bob", "department": "Engineering"},
        {"name": "Charlie", "department": "Finance"}
    ]
    with open(file_name, "w") as file:
        for employee in employees:
            line = f"Name: {employee['name']}"
            department = {employee['department']}
            "n"
            file.write(line)

```

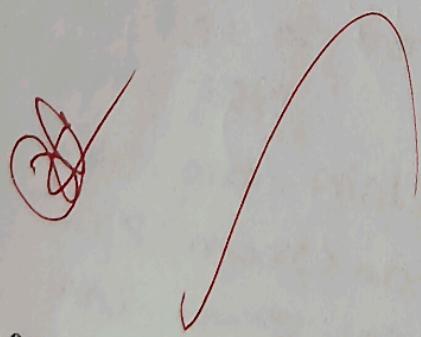
Name: Alice, Department: HR

Name: Bob, Department: Engineering

Name: Charlie, Department: Finance.

# Example usage:

```
write_employee_report("employee-report.txt")
```



Result: Thus the Python program implements various file operations was successfully executed and the output was verified.

VEL TECH	
LX No.	8
PERFORMANCE (5)	6
RESULT AND ANALYSIS (5)	6
VIVA VOCE (5)	3
RECORD (5)	2
TOTAL (20)	16
WITH DATE	