

Task 8.1 - Implement various text file operation.

15/9/25 (8.1).

Aim: to write a python program implement various text file operations.

Algorithm:

1. write to file:
 - Define write file (filename) function
 - open a file named "log.txt" in write mode.
 - write the following text to file.
 - Error objects are thrown when runtime errors occur. The object can also be base object for user-defined
 - close the file
2. Read from a file:
 - Define readfile (filename) function.
 - open the file specified by filename in read mode using a with statement.
 - Read the entire content of file.
 - print content.
3. execute program:
 - call writefile (write) to write predefined text to log.txt.
 - call readfile ("text") to attempt to read a file named "text" and print its content.
 - call readfile (

program 8.1

```
def writefile (filename):  
    f = open ("log.txt", "w")  
    f.write ("Error objects are thrown when runtime error  
    occur. the error object can also be used base object user-defined  
    exceptions")  
    f.close()  
def readfile (file name):  
    with open (file name, "r") as f:
```


(throw) throw
 (log) log
 (error) error
 output error object are thrown when runtime errors occur. The error object can also be used as a base object for user-defined exceptions.

VIT TECH - CSE

EX NO	
PERFORMANCE (%)	
RESULT AND ANALYSIS (%)	
VIVA VOCE (%)	
RECORD (%)	
TOTAL (%)	

H1: I am created by a function passed as an argument
 H2: I am created by a function passed as an argument

I am created by a function passed as an argument
 I am created by a function passed as an argument
 I am created by a function passed as an argument
 I am created by a function passed as an argument

Task 8.1 - Implement various file operations

Hint: To write a Python program implement various file operations

Algorithm

1. write to file:

• Before write file (filename) function:

output

Number of lines with 'ERROR' is 2.

• open a file named "log.txt" in write mode
• write the following text to the file

• Close the file

2. Read from file:

• Before readable (filename) function

• open the file specified by filename in read mode
• read the entire content of the

• print content

3. Create programs

• will writeable (write) to write predefined text to log.txt

• Call readable ("text") to attempt to read or file name
"text" and print its content

• will readable (

Program 8.1

def writeable (filename):

f = open ("log.txt", "w")

f.write ("Error objects are thrown when runtime error

error. The error object can also be used for user-defined

exceptions")

f.close()

def readable (filename):

with open (filename, "r") as f:

print (Content)

write file ("write")

read file ("text")

Problem 8.2:-

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

Algorithm:

1. Initialize Error Counter:

- Define the function count-error-lines (filename):
- Initialize error-count to 0.

2. open and Read file:

open the file specified by filename in read using a with statement.

3. check each line for "error".

• loop through each line in file.

• If the line contains the word "ERROR", Increment error-count by 1.

4. Return Error count:

• After reading all lines, return value of error-count

5. execute program:

• call count-error-lines ("log.txt") to count the number of line with the word "error" in file "log.txt".

• print result with message. "Number of lines with error"

Program :-

```
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("log.txt")
```

```
print(f"Number of lines with 'error' is {error-lines}")
```


output

Name: Alice, Department: HR.

Name: Bob, Department: Engineering

Name: Charlie, Department: Finance

log.txt

"Error" objects are thrown when runtime error occurs.

The Error object can be used as base object for user-defined exceptions.

Problem 8.3:-

You need to write report containing the details of employee in list. Write python that write this report to file named Employee-report.txt.

My 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 employees.
2. open file for writing:
 - open the file specified by filename write_employee_report.
3. write employee data to file:
 - loop through each employee in employee list:
 - for each employee, format a string as "Name": employee["Name"].
 - write the formatted string to file, followed by a newline character (\n).
4. Execute the program:
 - call write_employee_report("Employee-report.txt") to write employee data to file "employee-report.txt".

program

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


file.write (Name).

example usage.

write - employee - report ("employee.txt").

1. Initialize the list of grades

grades = [85, 90, 78, 92, 87]

display the grades list

print("grades list", grades)

prompt the user to enter index of grade they want to display

index = int(input("Enter the index of grade you want to display"))

attempt to display the grade at specified index

print(f"Grade at index {index} is {grades[index]}")

except IndexError:

handle the error

print("Invalid index")

except ValueError:

VEL TECH - CSE	
EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGNATURE	8

Result? Thus, the python program implement various text file operation was successfully and output was verified.