

Task-6: Implement various text file operation

Aim: To code a python program implement various text file operation.

Program 6.1:

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

Algorithm

1) Write a file:

- Define writefile(file name) function:-
- open a file named "log.txt" in write mode.
- write the following txt to the file:
 - Error objects are thrown when run times errors occur. the error object can also be used as a base object for user defined exceptions
- close the file

2) Read from a file:

- Define readfile(file name) 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 read file ("text") to attempt to read from a file named "text" and print its content

Program 6.1

```
def write_file(filename):  
    f = open('log.txt', 'w')  
    f.write("Error objects are thrown when  
    run time error 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)  
        write_file("write")  
        read_file("text")
```

↳ Exception is a class of objects which are used to handle errors.

↳ try block contains the statements which may cause error. If any error occurs then the control goes to the except block. If no error occurs then the control goes to the finally block.

Output:

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

↳ try block contains the statements which may cause error. If any error occurs then the control goes to the except block. If no error occurs then the control goes to the finally block.

↳ finally block contains the statements which are executed irrespective of whether an exception occurs or not.

6.2 Program

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

Algorithm

1) Initialize Error count:

- define the function count-error-line (file-name);
- initialize error-count to 0

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 num be the value of "Error" in the file "log.txt"
- print the result with the message: "Number of lines with 'ERROR': <error-lines>"

Program 6.2

```
def count_error_lines(file name):
    error_count = 0
    with open(file name, "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("Number of lines with 'ERROR':", error_lines)
log.txt
```

"Error objects are thrown when run time errors occur".

The Error object can also be used as a base object for a user defined exception".

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

Program 6.3

1) Create employee data:

- Define the function write_employee_report(file name):
- 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 write statement

“HORAS” និង សំណើរាយ
សារព័ត៌មាន និង របៀប ការងារ នៃ
សាខាបច្ចុប្បន្ន និង សាខាបច្ចុប្បន្ន និង
សាខាបច្ចុប្បន្ន និង សាខាបច្ចុប្បន្ន និង

• Before the formation of coffee-egg-cell.

(SOMA-SIA)

• infinite possibilities of combining colors.

ad part: number of lines with '00000' is 2

• କାନ୍ଦିଲା ପାଇଁ ଏହା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା

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

- call write_employee_report ("employee-report.txt") to write the employee data to the file employee ("employee-report.txt")

Program 6.3

```
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(line)~~

Example usage:

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

classmate -> classmate

classmate = 0

classmate = 0

classmate = 0

classmate = 1

classmate = 1

classmate = 1

classmate = 1

outputs: marks obtained are displayed

name:- Alice, department: HR

name:- Bob, department: engineering

name:- Charlie department: finance

classmate = 0

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	15/20
SIGN WITH DATE	

Result:

Thus the ~~python~~ program implemented, various text file operation was successfully executed and the output was verified.