|  |  |
| --- | --- |
| **Ex. No. 14**  **Date:12.07.2021** | **FILES AND EXCEPTION** |

**AIM:**

To write a python program using files and exception.

**PROGRAMMING BASE:**

**FILES**

I. Opening a File

* open () function in Python is to open a file in read or write mode.
* This will return a file object.
* Syntax: open(filename, mode).
* Access Mode

“ r “, for reading.

“ w “, for writing.

“ a “, for appending.

“ r+ “, for both reading and writing

II. Reading a File

* read() : Returns the read bytes in form of a string. Reads n bytes, if no n specified, reads the entire file.

File\_object.read([n])

* readline() : Reads a line of the file and returns in form of a string.
* For specified n, reads at most n bytes.
* However, does not reads more than one line, even if n exceeds the length of the line.

File\_object.readline([n])

* readlines() : Reads all the lines and return them as each line a string element in a list.

File\_object.readlines()

III. Writing into File

* write() : Inserts the string str1 in a single line in the text file.

File\_object.write(str1)

* writelines() : For a list of string elements, each string is inserted in the text file.
* Used to insert multiple strings at a single time.

File\_object.writelines(L) for L = [str1, str2, str3]

IV. Closing a File

* close() function closes the file and frees the memory space acquired by that file.
* File\_object.close()

**EXCEPTION**

* An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions
* Syntax:

try:

# Suspected Statements

......................

exceptExceptionI:

# Handles the exception, if there is ExceptionI

exceptExceptionII:

# Handles the exception, if there is ExceptionII

......................

else:

# If there is no exception then execute this block.

**Finally Block**

try:

# Suspected Statements

......................

Due to any exception, this may be skipped.

finally:

# This would always be executed.

......................

**PROGRAMS:**

**a) Count number of words in a file**

**Description:**

Create a file with some content and count the number of words in the file.

Get the file name from user.

Use Exception handler to handle IOError.

Sample Input:

first.txt

Sample Output:

Number of words in the file is 25

Sample Input:

second.txt

Sample Output:

File not found

**Program:**

’’’Name: R.sridevi

Roll.no:20UIT021

Program name: To count the number of words in the file.’’’

with open("first.txt",'w') as wf:

wf.write('hi i am madhumitha welcome to the python programming')

#Get the file name from the user

try:

file\_name = input()

#open file in read mode

with open(file\_name) as f:

#read the content from the file

content=f.read()

content=content.split(' ')

#count the words

count=len(content)

#print count

print("Number of words in the file is",count)

except IOError:

print("File not found")

**Output:**

first.txt

Number of words in the file is 12

Second.txt

File not found

**b) Electricity Bill Generation using Files**

**Description:**

Create a file with

First line - Previous month reading

Second line - Current month reading

Electricity Bill Generation

===================

unit = current month reading - previous month reading

Condition:

  First 100 units - Rs. 1 per unit

  101-200 units - Rs. 2.50 per unit

  201 -500 units - Rs. 4 per unit

> 501 units - Rs. 6 per unit

Get the file name from user.

Use Exception handler to handle IOError.

Sample Input:

first.txt

Sample Output:

EB Bill amount is 894

Sample Input:

second.txt

Sample Output:

File not found

**Program:**

'''Name: R.Sridevi

Roll No:20UIT021

Program Electricity Bill Generation using Files'''

with open("first.txt",'w') as wf:

wf.write('300 500')

#Get the file name from the user

try:

file\_name = input()

#open file in read mode

with open(file\_name) as f:

#read the content from the file

content=f.read()

content=content.split(' ')

#convert all readings into integer

for i in range(len(content)):

content[i]=int(content[i])

#find how many unit

unit=content[1]-content[0]

#find per unit cost

if(unit<=100):

per\_unit=1

elif unit<=200:

per\_unit=2.50

elif unit<=500:

per\_unit=4

else:

per\_unit=6

#print bill amount

print("EB Bill amount is",int(unit\*per\_unit))

except IOError:

print("File not found")

**Output:**

First.txt

EB Bill amount is 1920

**RESULT:**

Thus, the Python programs are executed successfully.