LAB # 15

FILE PROCESSING

OBJECTIVE

To explore methods to access, open/close , modes contained in external files.

THEORY

When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.

File Handling in Python

- ✓ File handling refers to the process of performing operations on a file, such as creating, opening, reading, writing and closing it through a programming interface.
- ✓ It involves managing the data flow between the program and the file system on the storage device, ensuring that data is handled safely and efficiently.
- ✓ Python allows users to handle files to read and write files, along with many other file handling options.
- ✓ Python treats file differently as **text** or **binary** and this is important.
- ✓ Each line of code includes a sequence of characters and they form text file.
- ✓ Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character.
- ✓ It ends the current line and tells the interpreter a new one has begun..

Importance of File Handling

File handling helps in managing large amounts of data efficiently, supports automation tasks, and ensures safe interaction between the program and the storage system.

File Operations

There are different operations that can be carried out on a file, these are:

- Creation of a new file
- Opening an existing file
- Reading from a file
- Writing to a file
- Closing a file

LAB # 15

1. Creating a file:

To create a file**, Python uses the Open() function** in Python to open a file in read or write mode.

syntax: open(Directory:\\filename, mode)
file = open('filename.txt', 'mode')

- **filename.txt:** name (or path) of the file to be opened.
- **mode:** mode in which you want to open the file (read, write, append, etc.).

There are various kinds of mode, that python provides:

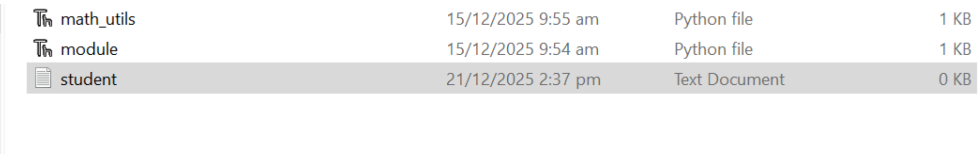
Mode	Description
'r'	Open a file for reading. (default)
'w'	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)
'rb'	Opens a file for reading binary data.
'wb'	Opens a file for writing binary data.

Example: Let's create a file and close it.

```
# Creating a file with open() function
# Open("filename", "mode")

file = open("student.txt", "w")
file.close()
```

File is created with student name in same directory:



math_utils	15/12/2025 9:55 am	Python file	1 KB
module	15/12/2025 9:54 am	Python file	1 KB
student	21/12/2025 2:37 pm	Text Document	0 KB

LAB # 15

2. Writing Data into a File:

Write() function, to manipulate the file, write the following in your Python environment:

syntax: file.write()

The close() command terminates all the resources in use and frees the system of this particular program.

syntax: file.close()

Example: Let's write into a Sudent file

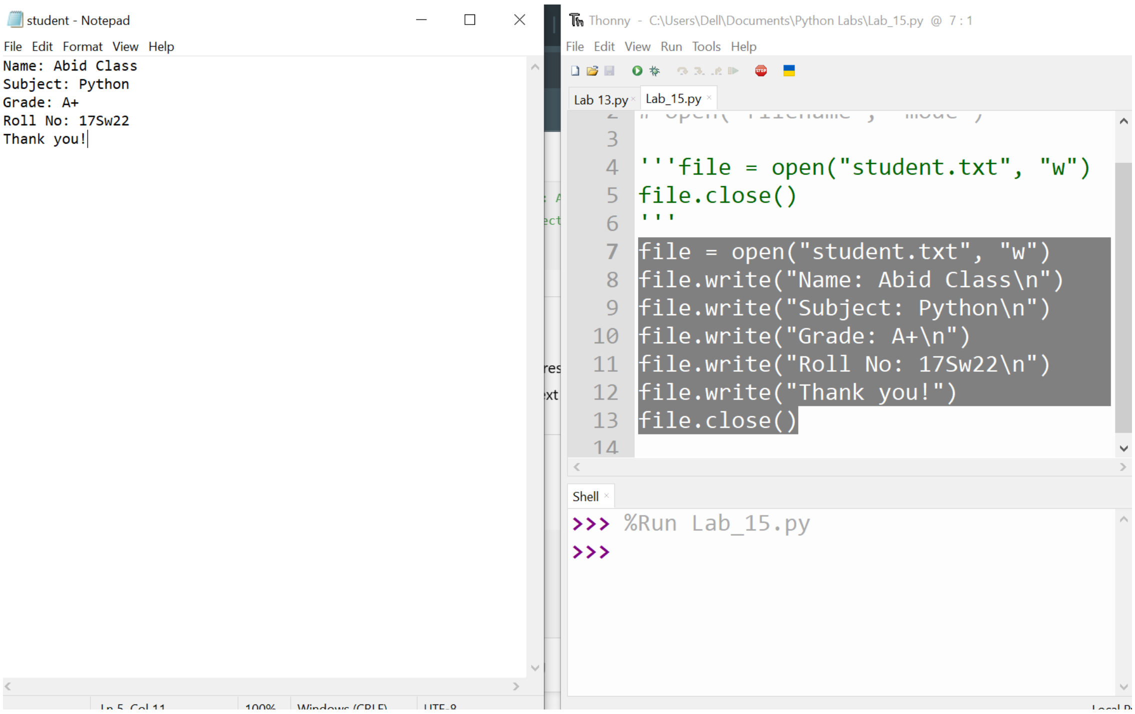
```
# Let's open the file and write into it.

file = open("student.txt", "w")

file.write("Name: Abid Class\n")
file.write("Subject: Python\n")
file.write("Grade: A+\n")
file.write("Roll No: 17Sw22\n")
file.write("Thank you!")

file.close()
```

Output



student - Notepad
File Edit Format View Help
Thonny - C:\Users\Dell\Documents\Python Labs\Lab_15.py @ 7 : 1
File Edit View Run Tools Help
Lab 13.py
Lab_15.py
Shell
Ln 5, Col 11
100%
Windows (CRLF)
UTF-8

```
Name: Abid Class
Subject: Python
Grade: A+
Roll No: 17Sw22
Thank you!
```

```
# open( filename , mode )

'''file = open("student.txt", "w")
file.close()
'''
file = open("student.txt", "w")
file.write("Name: Abid Class\n")
file.write("Subject: Python\n")
file.write("Grade: A+\n")
file.write("Roll No: 17Sw22\n")
file.write("Thank you!")
file.close()
```

```
>>> %Run Lab_15.py
>>>
```

Writing Data into a File Using **with Statement**

- ✓ The with statement is a safe and automatic way to work with files.
- ✓ It opens the file, allows writing, and **closes the file automatically.**

LAB # 15

If you will use this syntax: **file = open("student.txt", "w"),** it will **overwrite** the file, deleting the old content. If you want to **keep the existing content** and **add new content at the end**, you should use **append mode ("a")**:

Example: Let's write into a Sudent file using **with Statement**

```
# Using with statement with append

with open("Student.txt", "a") as file:
   file.write("\n")  # Adds an empty line between entries
   file.write("Name: Gh DPO\n")
   file.write("Subject: Java\n")
   file.write("Grade: A++\n")
   file.write("Roll No: 17Sw23\n")
   file.write('Thank you for using "with" and "append"!')
   file.write("\n")  # Adds an empty line between entries

with open("Student.txt", "r") as f:
   content = f.read()
   print(content)
```

Output

```
>>> %Run Lab_15.py

=== Old Content ===

Name: Abid Class
Subject: Python
Grade: A+
Roll No: 17Sw22
Thank you!

=== New Content Using with and append ===

Name: Gh DPO
Subject: Java
Grade: A++
Roll No: 17Sw23
Thank you for using "with" and "append"!
```

3. Reading a File Data

Read() mode: There is more than one way to read a file in Python. To extract a string that contains all characters in the file then it can be use as;

syntax: file.read()

LAB # 15

Example

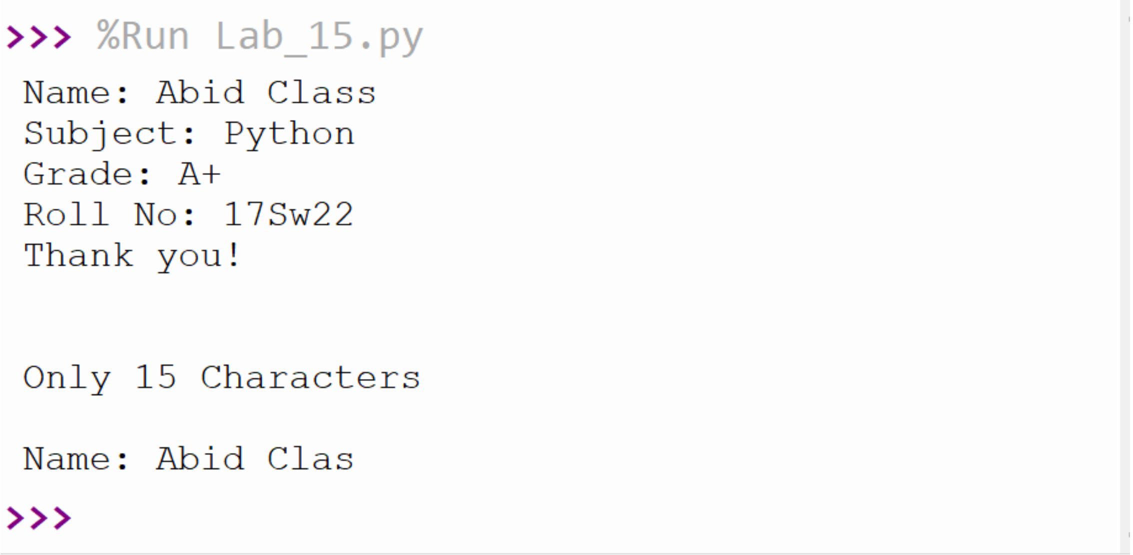
```
# Method – 01

file = open ("Student.txt", "r")
data = file.read()
print(data)
file.close()

# Method – 01
# Only starting character will be read and shown on screen.

file = open ("Student.txt", "r")
print(file.read(15))
file.close()
```

Output



```
>>> %Run Lab_15.py
 Name: Abid Class
 Subject: Python
 Grade: A+
 Roll No: 17Sw22
 Thank you!


 Only 15 Characters

 Name: Abid Clas
>>>
```

Checking File Properties

Example

```
# Checking the properties

f = open("Student.txt", "r")
print("Filename:", f.name)
print("Mode:", f.mode)
print("Is Closed?", f.closed)
f.close()
print("Is Closed?", f.closed)
```

LAB # 15

Output

```
>>> %Run Lab_15.py

Filename: Student.txt
Mode: r
Is Closed? False
Is Closed? True

>>>
```

Delete a File

To delete a file, you must import the OS module, and run its os.remove() function:

Example

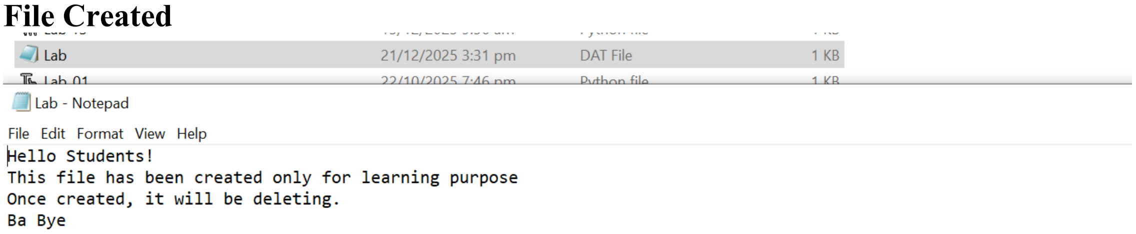
```
import os

with open("Lab.dat", "wb") as f:
    f.write(b"Hello Students!\n")
    f.write(b"This file has been created only for learning purpose\n")
    f.write(b"Once created, it will be deleting.\n")
    f.write(b"Ba Bye\n")

# Error Handling
try:
    os.remove("Lab.dat")
    print("File deleted successfully!")

except FileNotFoundError:
    print("File does not exist.")
```

File Created



```
Hello Students!
This file has been created only for learning purpose
Once created, it will be deleting.
Ba Bye
```

Output

```
>>> %Run Lab_15.py
Binary file deleted successfully!
>>>
```

LAB # 15

EXERCISE

A. Point out the errors, if any, in the following Python programs.

1. Code

```
file = open('python.txt', 'r')
print("Using read() mode character wise:")
s1 = file.read(19)
print(s1)
```

Output:

2. Code

```
f1 = open("jj",)
f1.write("something")
```

Output:

B. Creat a text file named python, Write the following code. Execute it and show the output. (You can use the Snipping Tool to take a snapshot of your txt.file)

1. Code

```
def main():
    # Open file for writing
    outfile = open('D:\\python.txt', 'w')
    # Write data to the file
    outfile.write("Bill Clinton\n")
    outfile.write("George Bush\n")
    outfile.write("Barack Obama\n")
    # Print file object (optional)
    print(outfile)
    # Close the output file
    outfile.close()
main()
```

LAB # 15

Output:

2. Code

```
def main():
    # Open file for exclusive creation
    outfile = open('D:\\python.txt', 'x')
    # Write data to the file
    outfile.write("var, account_balance, client_name\n")
    outfile.write("var = 1\naccount_balance = 1000.0\nclient_name = 'John Doe'\n")
    print(outfile)
    # Close the output file
    outfile.close()

main()
```

Output:

C. Write Python programs for the following:

1. Write a program that create a function called “file_read”, to read an entire text file.

2. Write a program that reads the content and replace any word in a string with different word.

Example:
Replace ‘dog’ with ‘cat’in a sentence

3. Write a program that prompts the user for their name. When they respond, write their name to a file called guest.txt.