FILE HANDLING – HANDWRITTEN STYLE NOTES

Date : ___________ Subject : Python Programming

Topic : File Handling in Python

→ File Handling means working with files (reading + writing)
using Python programs. It is used for storing data
permanently on the disk.

→ Python provides built-in functions for creating,
reading, writing, updating and deleting files.

TYPES OF FILES

----

1) Text Files (.txt)

- Human readable characters
- Example: "notes.txt", "data.txt"

2) Binary Files (.bin)

- Machine readable
- Example: images, videos, audio, excel etc.

STEPS IN FILE HANDLING

----

1) Open the File
2) Read or Write the File
3) Close the File

FILE OPENING MODES

"r" → Read mode (file must exist)

"w" → Write mode (overwrites file)

"a" → Append mode (adds data at end)

"r+" → Read + Write (no overwrite)

"w+" → Write + Read (overwrites)

"a+" → Append + Read

"x" → Create new file (error if already exists)

Binary Modes:

"rb" → Read binary

"wb" → Write binary

"ab" → Append binary

OPENING A FILE

----

f = open("data.txt", "r")

CLOSING A FILE

----

f.close()

IMPORTANT FILE METHODS

→ Reading:

f.read() – reads whole file

f.read(n) – reads n characters

f.readline() – reads one line

f.readlines() – returns list of lines

→ Writing:

f.write("text")

f.writelines(list_of_lines)

USING 'with' STATEMENT

----

with open("data.txt", "r") as f:

data = f.read()

→ No need to close file manually

→ Best practice

APPENDING DATA

----

with open("file.txt", "a") as f:

f.write("\nNew line added")

FILE POINTER METHODS

----

tell() → shows current pointer position

seek() → moves pointer to specific position

Example:

f.seek(0) → moves pointer to start of file

EXCEPTION HANDLING (IMPORTANT)

```
try:
f = open("abc.txt", "r")
except FileNotFoundError:
print("File not found")
```

WORKING WITH BINARY FILES

```
with open("img.jpg", "rb") as f:
data = f.read()

with open("copy.jpg", "wb") as f:
f.write(data)
```

CSV FILE HANDLING

```
import csv
```

Writing:

```
with open("data.csv","w",newline="") as f:
writer = csv.writer(f)
writer.writerow(["Name","Age"])
```

Reading:

```
with open("data.csv","r") as f:
```

reader = csv.reader(f)

for row in reader:

print(row)

JSON FILE HANDLING

----

import json

Writing:

json.dump(data, file)

Reading:

data = json.load(file)

FILE OPERATIONS USING OS MODULE

→ Check if file exists:

os.path.exists("file.txt")

→ Delete file:

os.remove("file.txt")

→ Rename file:

os.rename("old.txt","new.txt")

ADVANTAGES OF FILE HANDLING

✔ Stores data permanently

✔ Easy to manage large data

✔ Supports text, binary, csv, json formats

DISADVANTAGES

✘ Requires error handling

✘ Slow compared to RAM operations

✘ Risk of overwriting files

REAL WORLD USES

→ Banking systems

→ Student records

→ Logging system

→ E-commerce orders storage

→ Data analysis

END OF NOTES