Day 9 of 6 weeks Python course:

Time Breakdown (2 Hours)

Time Slot	Task
0:00 - 0:20	Introduction to File Handling
0:20 - 0:40	Reading Files (`read()`, `readlines()`)
0:40 - 1:00	Writing and Appending to Files
1:00 - 1:20	Deleting Files & Error Handling
1:20 - 2:00	Mini-Project: Notes Manager

1️⃣ What is File Handling? ✅ Why Use Files? Store persistent data (e.g., user data, logs, notes). Retrieve and update data without losing it when the program exits. ✅ Opening a File (open())

In [ ]:
```
file = open("sample.txt", "r")  # "r" means read mode
content = file.read()
print(content)
file.close()
```

Modes in `open()`:

Mode	Meaning
`r`	Read (default)
`w`	Write (Overwrites if file exists)
`a`	Append (Adds new content)
`x`	Create a new file (Fails if exists)

2️⃣ Reading from Files (read(), readlines()) ✅ Reading Entire File (read())

In [ ]:
```
file = open("sample.txt", "r")
print(file.read())  # Reads the entire file content
file.close()
```

✅ Reading Line by Line (readline())

In [ ]:
```
file = open("sample.txt", "r")
print(file.readline())  # Reads first line
file.close()
```

✅ Reading All Lines into a List (readlines())

In [ ]:
```
file = open("sample.txt", "r")
lines = file.readlines()  # Stores lines as a list
for line in lines:
    print(line.strip())  # strip() removes extra spaces/newlines
file.close()
```

✅ Using with open() for Auto-Closing

In [ ]:
```
with open("sample.txt", "r") as file:
    content = file.read()
```

```
    print(content)
```

3️⃣ Writing and Appending to Files ✅ Writing to a File (w mode - Overwrites the file)

In [ ]:
```
with open("sample.txt", "w") as file:
    file.write("Hello, Deepak!\n")
    file.write("Python is awesome!\n")
```

💡 Note: This overwrites existing content! ✅ Appending to a File (a mode - Adds new content)

In [ ]:
```
with open("sample.txt", "a") as file:
    file.write("Let's learn file handling!\n")
```

4️⃣ Deleting Files & Error Handling ✅ Deleting a File

In [ ]:
```
import os
os.remove("sample.txt")  # Deletes the file
```

✅ Checking if a File Exists Before Deleting

In [ ]:
```
if os.path.exists("sample.txt"):
    os.remove("sample.txt")
else:
    print("File does not exist.")
```

✅ Handling File Errors

In [ ]:
```
try:
    with open("nonexistent.txt", "r") as file:
        print(file.read())
except FileNotFoundError:
    print("File not found!")
```

🎯 Mini-Project: Notes Manager (Save & Retrieve Notes) 📌 Project Goal 1.Save notes in a file. 2.Allow users to view, add, delete notes. 3.Use file handling for persistence. 💻 Code Implementation

In [ ]:
```
import os

# File name for storing notes
filename = "notes.txt"

# Function to add a note
def add_note():
    note = input("Enter your note: ")
    with open(filename, "a") as file:
        file.write(note + "\n")
    print("Note added successfully!\n")

# Function to view all notes
def view_notes():
    if os.path.exists(filename):
        with open(filename, "r") as file:
            notes = file.readlines()
            if notes:
                print("\n📌 Your Notes 📌")
                for i, note in enumerate(notes, start=1):
                    print(f"{i}. {note.strip()}")
            else:
                print("No notes found.\n")
    else:
```

```
        print("No notes file found.\n")

# Function to delete a note
def delete_note():
    view_notes()
    if os.path.exists(filename):
        with open(filename, "r") as file:
            notes = file.readlines()

        if notes:
            try:
                note_no = int(input("Enter the note number to delete: ")) - 1
                if 0 <= note_no < len(notes):
                    deleted = notes.pop(note_no)
                    with open(filename, "w") as file:
                        file.writelines(notes)
                    print(f"Deleted note: {deleted.strip()}\n")
                else:
                    print("Invalid note number!\n")
            except ValueError:
                print("Invalid input! Please enter a number.\n")
        else:
            print("No notes to delete.\n")

# Main Menu
while True:
    print("\n📝 Notes Manager 📝")
    print("1. Add Note")
    print("2. View Notes")
    print("3. Delete Note")
    print("4. Exit")

    choice = input("Enter your choice (1-4): ")

    if choice == "1":
        add_note()
    elif choice == "2":
        view_notes()
    elif choice == "3":
        delete_note()
    elif choice == "4":
        print("Exiting Notes Manager. Goodbye!")
        break
    else:
        print("Invalid choice! Please enter a valid option.\n")
```

🔍 Step-by-Step Explanation

📌 Step 1: Store Notes in a File

python Copy Edit

```python
filename = "notes.txt"
```

- The **file** stores all notes.

📌 Step 2: Adding a Note

python Copy Edit

```python
with open(filename, "a") as file:
    file.write(note + "\n")
```

- **Appends notes** without overwriting existing ones.

📌 Step 3: Viewing Notes

python Copy Edit

```python
with open(filename, "r") as file:
    notes = file.readlines()
```

- **Reads all notes** and **prints them**.

📌 Step 4: Deleting a Note

python Copy Edit

```python
note_no = int(input("Enter the note number to delete: ")) - 1
notes.pop(note_no)
with open(filename, "w") as file:
    file.writelines(notes)
```

- **Removes a selected note** and updates the file.

✅ **Example Output:**

mathematica Copy Edit

```
📝 Notes Manager 📝
1. Add Note
2. View Notes
3. Delete Note
4. Exit

Enter your choice (1-4): 1
Enter your note: Learn Python File Handling
Note added successfully!
```

In [ ]:

```
📌 Summary of Day 9
✔ Learned File Handling (open(), read(), write())
✔ Practiced Error Handling & Deleting Files
✔ Completed a Mini-Project: Notes Manager
```