📓 Lecture 14: File Handling, JSON & Exception Handling

🗀 What is File Handling?

File Handling = 🗀 Working with external files (read/write/append)

🧠 Why?

- Save data permanently (outside code)
- Share logs or datasets
- Input/output with users
- Persist data after execution

🔧 File Handling Syntax

```
python
CopyEdit
file = open("filename.txt", "mode")
```

Mode	Purpose
r	Read (default)
w	Write (overwrite)
a	Append (add end)
r+	Read & write

✅ **Close file** after use:

```
python
CopyEdit
file.close()
```

📌 Better:

```
python
CopyEdit
with open("file.txt", "r") as file:
    ...
```

📝 Examples

✅ Create + Write:

```
python
CopyEdit
with open('example.txt', 'w') as file:
    file.write("This is written using write mode.")
```

📖 Read File:

```
python
CopyEdit
try:
    with open('example.txt', 'r') as file:
        content = file.read()
    print(content)
except FileNotFoundError:
    print("File not found!")
```

➕ Append:

```
python
CopyEdit
with open('example.txt', 'a') as file:
    file.write("\nNew line appended.")
```

🔁 Read + Write:

```
python
CopyEdit
with open('example.txt', 'r+') as file:
    file.write("\nAdded with r+ mode.")
```

🌐 Working with JSON

JSON = JavaScript Object Notation
✅ Structured
✅ Platform-independent
✅ API-friendly

📄 Write & Read JSON:

```
python
CopyEdit
import json

data = {"name": "Haris", "age": 25}

# Save JSON
with open("data.json", 'w') as file:
    json.dump(data, file)

# Read JSON
with open("data.json", 'r') as file:
    loaded = json.load(file)
    print(loaded)
```

🛡 Exception Handling

Handles runtime errors without crashing program

Types of Errors

Error Type	Example
Syntax Error	`print("Hello` ❌ (missing `)
Runtime Error	`10/0` ❌ (ZeroDivisionError)

✅ Try-Except

```
python
CopyEdit
try:
    num = int(input("Enter a number: "))
    print(10 / num)
except ZeroDivisionError:
    print("❗ Cannot divide by 0!")
```

🎯 Multiple Exceptions

```
python
CopyEdit
try:
    a = int(input("Num1: "))
    b = int(input("Num2: "))
    print(a / b)
except ZeroDivisionError:
    print("Error: Cannot divide by zero")
except ValueError:
    print("Error: Input must be a number")
```

🌀 Generic Exception

```
python
CopyEdit
try:
    num = int(input("Enter number: "))
    print(10 / num)
except Exception as e:
    print("Error occurred:", e)
```

🔚 Finally Block

```
python
CopyEdit
try:
    file = open("example.txt", "r")
    print(file.read())
except FileNotFoundError:
    print("File not found!")
finally:
    print("Closing file...")
    file.close()
```

⚙️ Custom Exception

```
python
CopyEdit
def withdraw(amount):
    if amount < 0:
        raise ValueError("Amount cannot be negative")
    print(f"Withdrawing ${amount}")

try:
    withdraw(-100)
except ValueError as e:
    print("Error:", e)
```

💡 Best Practices

✅ Use `with open()` to auto-close files
✅ Use `try-except` for safe coding
✅ Don't leave `except:` empty
✅ Add user-friendly error messages
✅ JSON = great for APIs & configs

✅ Summary

Topic	Key Use
File I/O	Save/load data, logs, datasets
JSON	APIs, config, structured storage
Exceptions	Crash-free execution, custom errors

📌 Next Lecture:

👉 Working with `.csv` files & creating **Custom Error Classes**