

PYTHON FILE HANDLING & FINAL PROJECT – COMPLETE CONCEPTS

File Handling – Reading and Writing Data to Files

Files allow programs to store and retrieve data permanently, beyond temporary memory. Working with files involves opening, reading or writing, and closing them properly.

```
# Opening and reading a textfile
file = open("notes.txt", "r")
content = file.read()
print(content)
file.close()
```

Modes:

"r" – read mode (default)

"w" – write mode (overwrites existing file)

"a" – append mode (adds new data)

"x" – create mode (fails if file exists)

Writing to a file

```
with open("output.txt", "w") as f:
    f.write("Learning Python File Handling!\n")
```

Using with open() automatically closes the file — it's the recommended practice.

Working with Different File Formats (txt, csv)

```
# Reading line by line from a textfile
with open("data.txt", "r") as f:
    for line in f:
        print(line.strip())
import csv
```

```
# Writing data to CSV
with open("students.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Name", "Age", "Course"])
    writer.writerow(["Ava", 20, "Python"])
```

```
# Reading data from CSV
with open("students.csv", "r") as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

Error Handling in File Operations

```
try:
    with open("unknown.txt", "r") as f:
        data = f.read()
except FileNotFoundError:
    print("File not found! Please check the file name.")
except PermissionError:
    print("You don't have permission to access this file.")
```

```
finally:
    print("Operation completed.")
```

Organizing Code into Multiple Files

```
# utils.py
def greet(name):
    print(f"Hello, {name}!")

# main.py
import utils
utils.greet("Liam")
```

Basic Debugging Techniques

Use `print()` statements, `assert` checks, or the `pdb` module for debugging.

```
import pdb
pdb.set_trace()  #sets a breakpoint for step-by-step execution
```

Final Project – Putting It All Together

Build a mini project using all concepts learned so far: Functions, Dictionaries, File Handling, Error Handling.

```
import csv

def add_student(name, age, course):
    with open("students.csv", "a", newline="") as f:
        writer = csv.writer(f)
        writer.writerow([name, age, course])
    print("Student added successfully!")

def view_students():
    try:
        with open("students.csv", "r") as f:
            reader = csv.reader(f)
            for row in reader:
                print(row)
    except FileNotFoundError:
        print("No data found. Please add students first.")

add_student("Ava", 20, "Python")
view_students()
```

1]
43s



```
# Load tasks
try:
    with open("todo.txt", "r") as f:
        tasks = f.read().splitlines()
except FileNotFoundError:
    tasks = []

while True:
    print("\n1. Add Task\n2. View Tasks\n3. Exit")
    choice = input("Choose: ")

    if choice == "1":
        t = input("Task: ")
        tasks.append(t)
        with open("todo.txt", "a") as f:
            f.write(t + "\n")
    elif choice == "2":
        print("\nYour Tasks:")
        for i, task in enumerate(tasks, start=1):
            print(i, "-", task)
    else:
        break
```

...

```
1. Add Task
2. View Tasks
3. Exit
Choose: 1
Task: Baking
```

```
1. Add Task
2. View Tasks
3. Exit
Choose: 1
Task: Cleaning
```

```
1. Add Task
2. View Tasks
3. Exit
Choose: 1
Task: Tax filing
```

```
1. Add Task
2. View Tasks
3. Exit
Choose: 2
```

```
Your Tasks:
1 - Baking
2 - Cleaning
3 - Tax filing
```

```
1. Add Task
2. View Tasks
3. Exit
Choose: 3
```

1]
4s



```
filename = input("Enter filename to read: ")

try:
    with open(filename, "r") as f:
        print(f.read())
```

```

filename = input("Enter filename to read: ")

try:
    with open(filename, "r") as f:
        print(f.read())
except FileNotFoundError:
    print("Error: File does not exist.")
except PermissionError:
    print("Error: No permission to read this file.")
except Exception as e:
    print("Unexpected error:", e)

```

```

Enter filename to read: todo.txt
Baking
Cleaning
Tax filing

```

```

import csv
from datetime import datetime

FILE = "expenses.csv"

def init_file():
    try:
        with open(FILE, "x") as f:
            writer = csv.writer(f)
            writer.writerow(["Date", "Category", "Amount"])
    except FileExistsError:
        pass # File already exists, no need to recreate

def add_expense():
    category = input("Enter category: ")
    amount = float(input("Enter amount: "))
    date = datetime.now().strftime("%Y-%m-%d")

    with open(FILE, "a", newline="") as f:
        writer = csv.writer(f)
        writer.writerow([date, category, amount])

    print("✅ Expense added successfully!")

def view_expenses():
    try:
        with open(FILE, "r") as f:
            reader = csv.reader(f)
            print("\n--- All Expenses ---")
            for row in reader:
                print(row)
    
```

```

except FileNotFoundError:
    print("No expense data found. Please add expenses first.")

def monthly_report():
    month = input("Enter month (YYYY-MM): ")
    total = 0

    try:
        with open(FILE, "r") as f:
            reader = csv.reader(f)
            next(reader) # skip header
            for date, category, amount in reader:
                if date.startswith(month):
                    total += float(amount)

        print(f"\n📊 Total expenses for {month}: ₹{total}")

    except:
        print("An error occurred while generating the report.")

if __name__ == "__main__":
    init_file()

    while True:
        print("\n=====")
        print(" PERSONAL FINANCE TRACKER ")
        print("=====")
        print("1. Add Expense")
        print("2. View All Expenses")
        print("3. Monthly Report (YYYY-MM)")
        print("4. Exit")

        choice = input("Enter choice: ")
    
```

```
2. View All Expenses
3. Monthly Report (YYYY-MM)
... 4. Exit
Enter choice: 1
Enter category: Accessories
Enter amount: 5000
✅ Expense added successfully!
```

```
=====
PERSONAL FINANCE TRACKER
=====
```

```
1. Add Expense
2. View All Expenses
3. Monthly Report (YYYY-MM)
4. Exit
Enter choice: 1
Enter category: Food and rent
Enter amount: 25000
✅ Expense added successfully!
```

```
=====
PERSONAL FINANCE TRACKER
=====
```

```
1. Add Expense
2. View All Expenses
3. Monthly Report (YYYY-MM)
4. Exit
Enter choice: 1
Enter category: Self Care
Enter amount: 20000
✅ Expense added successfully!
```

```
=====
PERSONAL FINANCE TRACKER
=====
```

```
1. Add Expense
2. View All Expenses
```

```
✅ Expense added successfully!
```

```
=====
PERSONAL FINANCE TRACKER
=====
```

```
1. Add Expense
2. View All Expenses
3. Monthly Report (YYYY-MM)
4. Exit
Enter choice: 2
```

```
--- All Expenses ---
['Date', 'Category', 'Amount']
['2025-11-10', 'Dress Shopping', '2000.0']
['2025-11-10', 'Accessories', '5000.0']
['2025-11-10', 'Food and rent', '25000.0']
['2025-11-10', 'Self Care', '20000.0']
['2025-11-10', 'Orphanage funds', '15000.0']
```

```
=====
PERSONAL FINANCE TRACKER
=====
```

```
1. Add Expense
2. View All Expenses
3. Monthly Report (YYYY-MM)
4. Exit
Enter choice: 4
Exiting... Goodbye!
```