

File Handling & GUI

1. File Handling – الملفات

Explanation:

Files are used to store data permanently on disk. Unlike RAM, files keep the data even after the program stops.

Basic operations:

- Open a file
- Read from a file
- Write to a file
- Append to a file
- Close a file
- Rename / Delete a file

Modes:

- "r" → read (قراءة)
- "w" → write (كتابة – يحذف المحتوى القديم)
- "a" → append (إضافة)
- "x" → create new file only (إنشاء)

Best practice:

```
with open("data.txt", "r") as f:  
    text = f.read()
```

The file closes automatically when the block ends.

2. Reading & Writing Files

Reading methods:

- `read(n)` → read *n* characters
- `read()` → read all content
- `readline()` → read one line
- `readlines()` → read all lines into a list

Writing methods:

- `write()`
- `writelines()`

3. File Utilities (os module)

```
import os
os.rename("old.txt", "new.txt")
os.remove("file.txt")
```

Always check if the file exists before removing.

4. CLI vs GUI

- **CLI:** Commands in terminal, fast, needs memorization.
- **GUI:** Windows, buttons, icons, easier for users, event-driven.

5. Tkinter Basics

Tkinter is the standard GUI library for Python.

Steps:

1. Import Tkinter: `from tkinter import *`
2. Create main window: `root = Tk()`
3. Add widgets (Label, Button, Entry, etc.)
4. Use layout managers: `pack()`, `grid()`, `place()`

5. Run event loop: `root.mainloop()`

Worked Example – Button + MessageBox:

```
from tkinter import Tk, Button, messagebox

def hello():
    messagebox.showinfo("Hi", "Hello, world")

root = Tk()
root.title("Test")
Button(root, text="Say Hi", command=hello).pack()
root.mainloop()
```

6. Widgets

- **Label**: display text or image
- **Button**: perform an action
- **Entry**: input text
- **Checkbutton**: toggle options
- **Radiobutton**: select one option

Geometry Managers:

- `pack()` → block style
- `grid()` → row/column table
- `place()` → specific coordinates

7. Small GUI Example

Idea: Entry field + Save button → append text to `notes.txt`.

```

from tkinter import *

def save():
    text = entry.get()
    with open("notes.txt", "a") as f:
        f.write(text + "\n")
    entry.delete(0, END)

root = Tk()
entry = Entry(root)
entry.pack()
Button(root, text="Save", command=save).pack()
root.mainloop()

```

Homework

1. **File:** From hours.txt, print each worker's ID, name, total hours, average hours/day, and the max day.
2. **File:** Write to students.txt, then rename it to students_final.txt, then delete it.
3. **GUI:** Complete the calculator example by adding subtraction, multiplication, and division (handle division by zero).
4. **GUI + File:** Build an app with Entry for username and Save button → append to users.txt and show confirmation.
5. **Challenge:** Build a GUI with 2 Entry fields (hours for day1 and day2) and a button that shows the sum in a Label.