

UID EX 3

Rama Thulasi G
230701259

Command Line Interface (CLI)

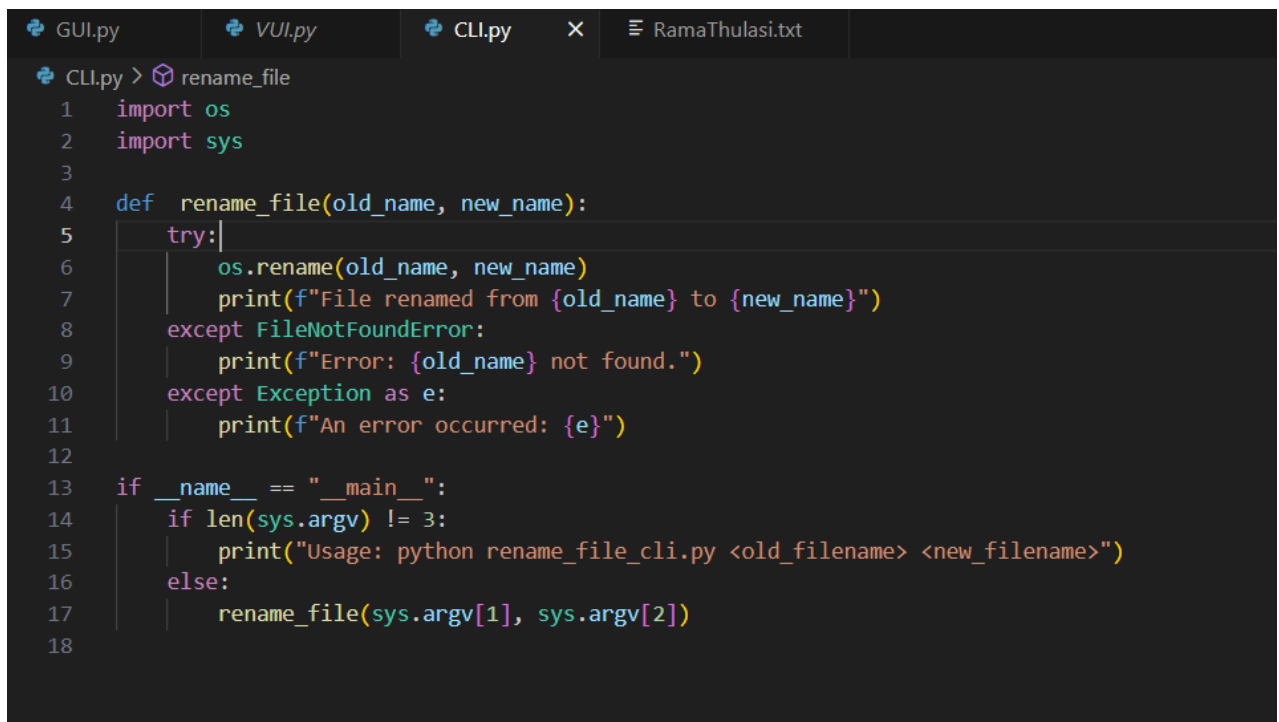
- A **Command Line Interface (CLI)** is a text-based interface where users interact with a computer by typing commands.
- It is commonly used by developers, system administrators, and advanced users to perform tasks efficiently.
- Very fast for experienced users who know commands.

Advantages:

- Faster for experienced users
- Uses fewer system resources

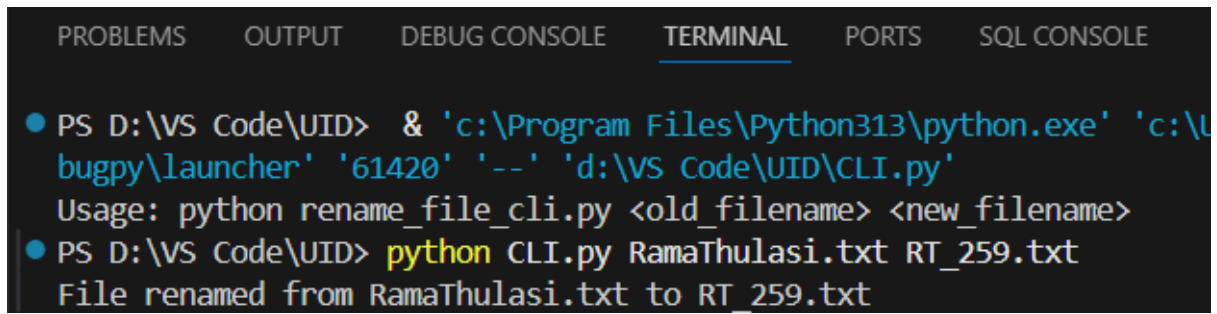
Disadvantages:

- Requires memorizing commands
- No visual elements



The screenshot shows a code editor with a dark theme. The top bar displays four tabs: 'GUI.py', 'VUI.py', 'CLI.py', and 'RamaThulasi.txt'. The 'CLI.py' tab is active. The editor content shows a Python script for renaming files. The script starts with line numbers 1 through 18. It imports 'os' and 'sys'. A function 'rename_file' is defined with parameters 'old_name' and 'new_name'. Inside the function, a 'try' block attempts to rename the file using 'os.rename'. If successful, it prints a confirmation message. It then has two 'except' blocks: one for 'FileNotFoundError' which prints an error message, and another for a general 'Exception' which prints a message about an error occurring. The script concludes with a standard 'if __name__ == "__main__":' block that checks the number of command-line arguments. If there are 3 arguments, it prints the usage instructions. Otherwise, it calls the 'rename_file' function with the provided arguments.

```
1  import os
2  import sys
3
4  def rename_file(old_name, new_name):
5      try:
6          os.rename(old_name, new_name)
7          print(f"File renamed from {old_name} to {new_name}")
8      except FileNotFoundError:
9          print(f"Error: {old_name} not found.")
10     except Exception as e:
11         print(f"An error occurred: {e}")
12
13 if __name__ == "__main__":
14     if len(sys.argv) != 3:
15         print("Usage: python rename_file_cli.py <old_filename> <new_filename>")
16     else:
17         rename_file(sys.argv[1], sys.argv[2])
18
```

A screenshot of a Visual Studio Code terminal window. The terminal has tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected and underlined), PORTS, and SQL CONSOLE. The terminal shows two commands being executed in a PowerShell prompt. The first command is a call to the Windows 'ren' command to rename a file. The second command is a call to a Python script 'CLI.py' to perform the same operation. The output of the second command shows the file was successfully renamed.

```
● PS D:\VS Code\UID> & 'c:\Program Files\Python313\python.exe' 'c:\U
bugpy\launcher' '61420' '--' 'd:\VS Code\UID\CLI.py'
Usage: python rename_file_cli.py <old_filename> <new_filename>
● PS D:\VS Code\UID> python CLI.py RamaThulasi.txt RT_259.txt
File renamed from RamaThulasi.txt to RT_259.txt
```

Graphical User Interface (GUI)

- A **Graphical User Interface (GUI)** allows users to interact with a computer using visual elements such as windows, buttons, icons, and menus.
- It is designed for ease of use, making computers accessible to non-technical users.

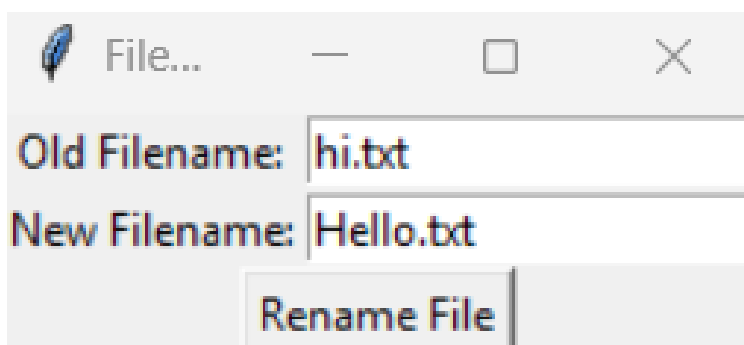
Advantages:

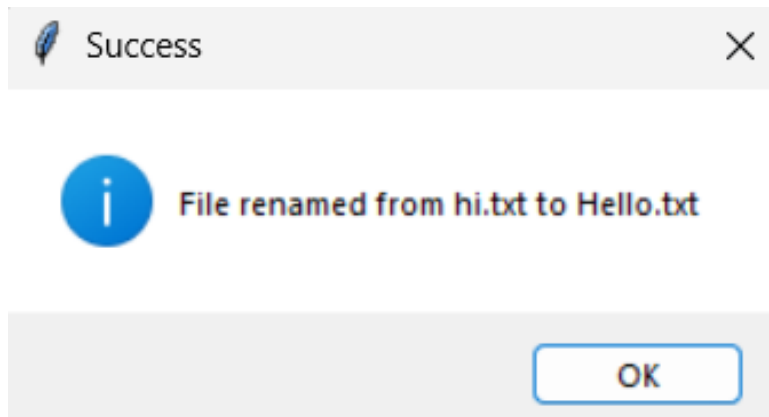
- Easy to use, beginner-friendly
- Does not require memorizing commands
- More interactive with visuals

Disadvantages:

- Slower than CLI for advanced users
- Requires more system resources
- Limited automation capabilities

```
GUI.py X VUI.py CLI.py RamaThulasi.txt
GUI.py > rename_file
1 import tkinter as tk
2 from tkinter import messagebox
3 import os
4
5 def rename_file():
6     old_name = old_filename_entry.get()
7     new_name = new_filename_entry.get()
8
9     try:
10         os.rename(old_name, new_name)
11         messagebox.showinfo("Success", f"File renamed from {old_name} to {new_name}")
12     except FileNotFoundError:
13         messagebox.showerror("Error", f"File {old_name} not found.")
14     except Exception as e:
15         messagebox.showerror("Error", f"An error occurred: {e}")
16
17 # Set up the main window
18 root = tk.Tk()
19 root.title("File Renamer")
20
21 # Create and place labels, entries, and buttons
22 tk.Label(root, text="Old Filename:").grid(row=0, column=0)
23 tk.Label(root, text="New Filename:").grid(row=1, column=0)
24
25 old_filename_entry = tk.Entry(root)
26 old_filename_entry.grid(row=0, column=1)
27
28 new_filename_entry = tk.Entry(root)
29 new_filename_entry.grid(row=1, column=1)
30
31 rename_button = tk.Button(root, text="Rename File", command=rename_file)
32 rename_button.grid(row=2, columnspan=2)
33
34 # Start the Tkinter event loop
35 root.mainloop()
36
```





Voice User Interface (VUI)

- A **Voice User Interface (VUI)** allows users to interact with a computer or device using spoken commands.
- It is commonly used in smart assistants, voice-controlled applications, and accessibility features.

Advantages:

- Hands-free and convenient
- Easy to use for all ages
- Useful for people with disabilities

Disadvantages:

- Accuracy depends on speech recognition
- Struggles with different accents and background noise
- Limited to predefined commands

```
GUI.py VUI.py X CLI.py RamaThulasi.txt
VUI.py > rename_file_from_voice_command
1 import speech_recognition as sr
2 import os
3
4 def rename_file_from_voice_command(command):
5     # Extracting old and new filename from the command
6     try:
7         words = command.split(" ")
8         old_name = words[1]
9         new_name = words[3]
10
11         os.rename(old_name, new_name)
12         print(f"File renamed from {old_name} to {new_name}")
13     except Exception as e:
14         print(f"Error: {e}")
15
16 def listen_for_command():
17     recognizer = sr.Recognizer()
18     mic = sr.Microphone()
19
20     print("Listening for command to rename a file...")
21     with mic as source:
22         recognizer.adjust_for_ambient_noise(source)
23         audio = recognizer.listen(source)
24
25     try:
26         command = recognizer.recognize_google(audio)
27         print(f"Command received: {command}")
28         rename_file_from_voice_command(command)
29     except sr.UnknownValueError:
30         print("Sorry, I couldn't understand the command.")
31     except sr.RequestError as e:
32         print(f"Could not request results from Google Speech Recognition service; {e}")
33
34 if __name__ == "__main__":
35     listen_for_command()
36
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
PS D:\VS Code\UID> & 'c:\Program Files\Python313\python.exe' 'c:\Users\bugpy\launcher' '61524' '--' 'd:\VS Code\UID\VUI.py'
Listening for command to rename a file...
Sorry, I couldn't understand the command.
PS D:\VS Code\UID>
```