Chapter 53 - Try Finally

Theory:

This chapter covers the topic of Try Finally.

Code Example:

```
# ==============================================
# ■ try with finally in Python — Simplified Explanation
# ==============================================
# ■ The finally block always runs — no matter what happens in the try block.
# ■ It's used for cleanup actions like closing a file, disconnecting from a server, etc.
# ■ Syntax:
# try:
# # risky code
# finally:
# # cleanup code that always runs
# ■ Think of it like:
# "Try to do this... No matter what happens — success, failure, or error — I'll do this
cleanup anyway."
# ----------------------------------------------
# ■ Example 1: Error in try block, but finally still runs
# ----------------------------------------------
try:
print("■ Opening file...")
x = 10 / 0 # ZeroDivisionError
finally:
print("■ File closed!") # always runs
# ----------------------------------------------
# ■ Example 2: File operations with finally
# ----------------------------------------------
try:
print("\n■ File opened.")
file = open("data.txt", "w")
print("■ Working on it...")
file.write("Hello!")
print("■ Text written to file!")
finally:
file.close()
print("■ File closed!")
# ----------------------------------------------
# ■ Example 3: User input with try-except-finally
# ----------------------------------------------
try:
a = int(input("\n■ Enter a number: "))
print(f"■ You entered: {a}")
except Exception as e:
print("■■ Error occurred!")
print(f"■ {e}")
finally:
print("■ I am inside the finally block. (Always runs)")
# ----------------------------------------------
# ■ Example 4: Using finally inside a function
# ----------------------------------------------
def main():
```

```
try:
a = int(input("\n■ Enter a number: "))
print(f"■ You entered: {a}")
except Exception as e:
print("■■ Error occurred inside function!")
print(f"■ {e}")
finally:
print("■ I am inside the finally block of the function.")
# ■ Call the function
main()
```