***EXCEPTION HANDLING***

Exception handling is used to manage runtime errors, ensuring the program doesn’t crash and allows recovery or graceful termination.

### ***Key Blocks:***

1. try Block: Contains the code that might raise an exception.
2. except Block: Handles specific exceptions that occur in the try block.
3. else Block (optional): Executes if no exception occurs in the try block.
4. finally Block (optional): Executes regardless of whether an exception occurred, often used for cleanup.

### ***Syntax:***

try:

# Code that might raise an exception

except ExceptionType1:

# Handle exception of type ExceptionType1

except ExceptionType2 as e:

# Handle exception of type ExceptionType2

# Optional: Use 'e' to access exception details

else:

# Code to execute if no exceptions occur (optional)

finally:

# Code that executes regardless of exception occurrence (optional)

### ***Example:***

try:

num = int(input("Enter a number: "))

result = 10 / num

except ZeroDivisionError:

print("Cannot divide by zero!")

except ValueError:

print("Invalid input! Enter a number.")

else:

print("Result:", result)

finally:

print("Execution complete.")

### ***Uses:***

1. Prevent program crashes.
2. Handle errors gracefully.
3. Log/debug unexpected issues.
4. Maintain application reliability.