PHP EXCEPTION HANDLING

What is Exception Handling?

When something goes wrong in a PHP program (like dividing by zero), instead of crashing, **exceptions** allow us to handle the error safely. PHP tells us, "Hey, there's a problem!" using **`try-catch` blocks**.

1. Basic Terms

- **`try` block**: Code that may throw an exception.
- **`throw`**: Creates an exception.
- **`catch` block**: Handles the exception.
- **`finally` block** (optional): Runs code after try-catch, whether or not an exception occurs.

How It Works: Step-by-Step

1. **`try` Block**:
    - The code inside `try` is **risky**.
    - If it runs without issues, great! But if something goes wrong, PHP **throws** an exception.
2. **`throw` Keyword**:
    - This **throws** an exception when the program detects a problem (like dividing by zero).
    - Think of it as saying: *"I found an error. Someone please handle it!"*
3. **`catch` Block**:
    - If an exception is thrown, the code inside `catch` **handles** it.
    - It prevents the program from crashing by displaying an error message.
4. **`finally` Block (Optional)**:
    - This block always runs, whether or not an error happens.
    - Useful for cleanup (like closing files or freeing resources).

```php
<?php
try {
   $x = 10;
   $y = 0;  // Division by zero
   if ($y == 0) {
      throw new Exception("You cannot divide by zero!");  // Throw an error
   }
   echo $x / $y;  // This line won't run if an exception is thrown
} catch (Exception $e) {
   echo "Caught Exception: " . $e->getMessage();  // Handle the error
} finally {
   echo "\nDone with error handling.";  // This always runs
}
?>
```

Breaking it Down Even More

- `$y = 0;`: Dividing by zero is not allowed, so we use `throw` to signal an issue.
- When PHP sees `throw`, it **jumps to the `catch` block**.
- Inside `catch`, we use `$e->getMessage()` to show the error message.
- Finally, the `finally` block runs to print "Done with error handling."

Output for the Above Code:

Caught Exception: You cannot divide by zero!

Done with error handling.

Why Is This Important?

- Without exception handling, PHP would crash when an error happens.
- With `try-catch`, we prevent crashes and handle errors safely.

Why Use Exception Handling?

- Makes error management cleaner.
- Allows multiple catch blocks for different exceptions.
- Improves code readability and prevents crashes