**Week 6: Exception Handling and Error Management in PHP**

**🟩 Slide 1: Learning Objectives**

After this lesson, you will be able to:

* Understand what exceptions and errors are in PHP
* Use try, catch, and finally for safer code
* Throw your own exceptions without using classes
* Apply best practices for error handling
* Debug and troubleshoot your PHP code efficiently

**🟦 Slide 2: What Is Exception Handling?**

* An **exception** is an unexpected event that interrupts normal program flow
* Example: file not found, division by zero, or database error

✅ PHP lets you **"catch"** errors before they crash your app

📌 **Goal:** Avoid displaying technical errors to the user and instead show helpful messages

**🟦 Slide 3: Basic Syntax – try, catch, finally**

php

CopyEdit

try {

// risky code here

} catch (Exception $e) {

// what to do if an error occurs

} finally {

// (optional) always runs after try/catch

}

* try — place the code that might cause an error
* catch — handles the error if it happens
* finally — runs always, even if no error

**🟦 Slide 4: Simple Example – Using try and catch**

php

CopyEdit

try {

$age = -1;

if ($age < 0) {

throw new Exception("Invalid age entered!");

}

echo "Age: $age";

} catch (Exception $e) {

echo "Error: " . $e->getMessage();

}

🖥️ **Output:**  
Error: Invalid age entered!

**🟦 Slide 5: try-catch-finally in action**

php

CopyEdit

try {

echo "Trying code...<br>";

throw new Exception("Oops!");

} catch (Exception $e) {

echo "Caught error: " . $e->getMessage() . "<br>";

} finally {

echo "Finished execution.<br>";

}

🖥️ **Output:**

nginx

CopyEdit

Trying code...

Caught error: Oops!

Finished execution.

✅ finally is useful for cleanup: closing files, disconnecting DB, etc.

**🟦 Slide 6: Throwing Exceptions Without Classes**

You can throw custom errors without defining a custom class:

php

CopyEdit

$score = 120;

try {

if ($score > 100) {

throw new Exception("Score cannot exceed 100.");

}

echo "Score: $score";

} catch (Exception $e) {

echo "Notice: " . $e->getMessage();

}

🖥️ Output: Notice: Score cannot exceed 100.

**🟦 Slide 7: Best Practices for Error Handling**

✅ Do’s:

* Use try and catch when dealing with risky operations (file I/O, input validation, DB)
* Show user-friendly messages
* Log detailed errors (in dev only)

❌ Don’ts:

* Don’t leave raw error messages on a live website
* Don’t overuse try/catch where not needed

**🟦 Slide 8: Debugging Techniques in PHP**

🧪 **Built-in Debug Tools:**

* var\_dump($variable) – shows type and value
* print\_r($array) – shows contents of arrays
* error\_log("Something went wrong!") – writes to log file

🧰 **Display all errors (for development only):**

php

CopyEdit

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);