

# Understanding Constructors and Destructors in PHP

# Learning Objectives

01

- Understand what a constructor is and why it is used

02

- Write PHP classes using `__construct()` to initialize data

03

- Understand the purpose of `__destruct()` and how it works

04

- Demonstrate how constructors and destructors run automatically

# What is a Constructor?

- A constructor is a special method that runs automatically when an object is created

- It is used to set initial values for the object's properties

## Example – Constructor in PHP

```
1  <?php
2
3  class Student {
4      public $name;
5      public $course;
6
7      public function __construct($name, $course) {
8          $this->name = $name;
9          $this->course = $course;
10     }
11
12     public function showInfo() {
13         return "Name: $this->name, Course: $this->course";
14     }
15 }
16
17 $student = new Student("Jerico", "BSIT");
18 echo $student->showInfo();
19
20
21 ?>
```

# What is a Destructor?

- A destructor is a special method that runs automatically when the object is destroyed

- Useful for cleanup tasks like:
- Closing a file
  - Closing a database
  - Goodbye message





## Example – Destructor in PHP

```
1  <?php
2  class Student {
3      public $name;
4      public $course;
5
6      public function __construct($name, $course) {
7          $this->name = $name;
8          $this->course = $course;
9          echo "Student object created.<br>";
10     }
11
12     public function showInfo() {
13         return "Name: $this->name, Course: $this->course";
14     }
15
16     public function __destruct() {
17         echo "<br>Student object destroyed.";
18     }
19 }
20 $student = new Student("Jerico", "BSIT");
21 echo $student->showInfo();
22 ?>
23
```

# What is a Method Overloading?

Creating multiple methods with the same name but different numbers or types of parameters

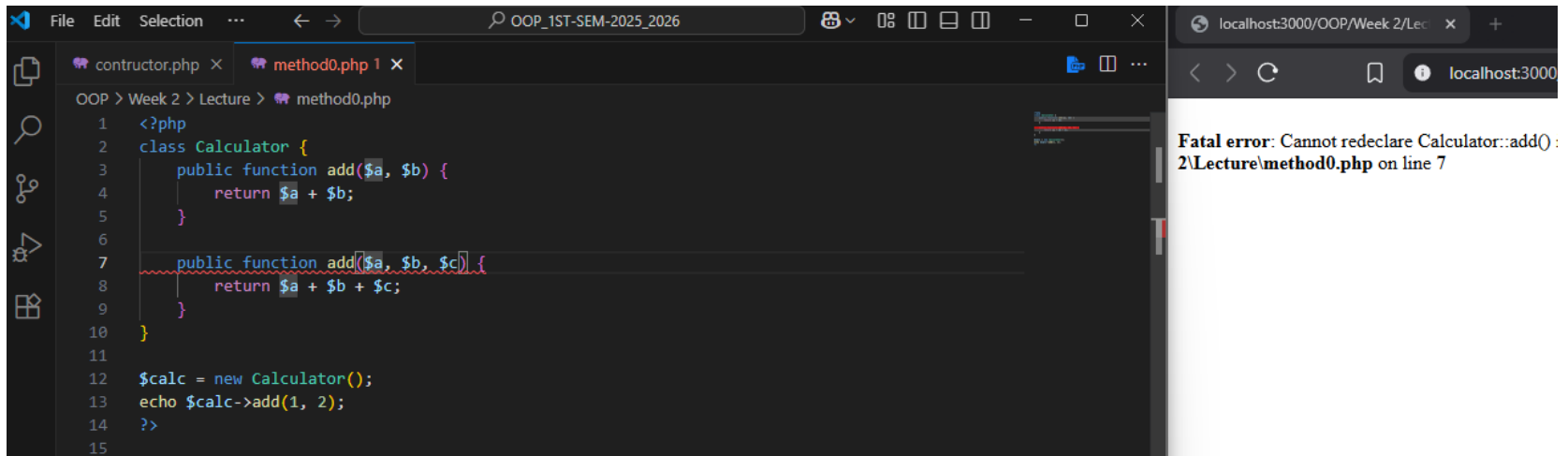
# Java Example: Method Overloading

Calculator.java	   Share 	Output
<pre>1 public class Calculator { 2 3     public int add(int a, int b) { 4         return a + b; 5     } 6 7     public int add(int a, int b, int c) { 8         return a + b + c; 9     } 10 11     public static void main(String[] args) { 12         Calculator calc = new Calculator(); 13 14         System.out.println("Sum of 2 numbers: " + calc.add(5, 3)); 15         System.out.println("Sum of 3 numbers: " + calc.add(5, 3, 2)); 16     } 17 } 18</pre>	<pre>Sum of 2 numbers: 8 Sum of 3 numbers: 10  === Code Execution Successful ===</pre>	



# PHP Doesn't Support It Directly

- PHP does not allow more than one method with the same name in a class
- If you try, it will give a fatal error



The screenshot shows a code editor with two files: `constructor.php` and `method0.php`. The `method0.php` file contains the following PHP code:

```
1 <?php
2 class Calculator {
3     public function add($a, $b) {
4         return $a + $b;
5     }
6
7     public function add($a, $b, $c) {
8         return $a + $b + $c;
9     }
10 }
11
12 $calc = new Calculator();
13 echo $calc->add(1, 2);
14 ?>
15
```

The code attempts to define two methods with the same name `add` in the `Calculator` class. The second method definition on line 7 is highlighted with a red squiggly line, indicating a syntax error.

To the right of the code editor, a browser window shows the error message:

```
Fatal error: Cannot redeclare Calculator::add() :
2\Lecture\method0.php on line 7
```

A decorative element on the left side of the slide consisting of three vertical bars of increasing height from left to right, colored in a dark purple shade.

# How PHP Handles It Instead:

- PHP uses a special magic method called `__call()` to simulate method overloading.
  - This method:
    - Automatically runs when you call a method that doesn't exist
  - Receives:
  - The method name as a string
  - The arguments in an array
  - You can then check how many arguments were passed and handle each case differently
-

# Method Overloading in PHP

```
contractor.php  method0.php X
OOP > Week 2 > Lecture > method0.php
1  <?php
2  class Calculator {
3      public function __call($method, $args) {
4          if ($method === "add") {
5              $count = count($args);
6
7              if ($count === 2) {
8                  return $args[0] + $args[1];
9              } elseif ($count === 3) {
10                 return $args[0] + $args[1] + $args[2];
11             } else {
12                 return "Invalid number of arguments for add()";
13             }
14         } else {
15             return "Method '$method' not found.";
16         }
17     }
18 }
19
20 $calc = new Calculator();
21
22 echo $calc->add(2, 3);
23 echo "<br>";
24 echo $calc->add(1, 2, 3);
25 echo "<br>";
26 echo $calc->add(5);
27 ?>
28
```

5  
6  
Invalid number of arguments for add()

A decorative element on the left side of the slide consisting of four vertical bars of increasing height from left to right, colored in a dark purple shade.

# SUMMARY

- Constructor is a special method that automatically sets initial values when an object is created.
  - Destructor is a special method that runs when the object is destroyed to clean up or finalize tasks.
  - Method overloading in PHP is simulated using `__call()` to handle method calls with different numbers of arguments.
-

# Hands-on Task

- Activity:
  - 1. Create a class 'Book'
  - 2. Use a constructor to set title and author
  - 3. Use a method to display the book info
  - 4. Use a destructor to say 'Book object destroyed.'
- 