**New Acknowledges**

**PHP OOP Constructor**

A **constructor** allows you to initialize an object's properties upon creation of the object. If you create a \_\_construct() function, **PHP** will automatically call this function when you create an object from a class.

**PHP Connect to MySQL**

PHP 5 and later can work with a MySQL database using:

* **MySQLi extension** (the "i" stands for improved)
* **PDO (PHP Data Objects)**

Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

**Should I Use MySQLi or PDO?**

If you need a short answer, it would be "Whatever you like".

Both MySQLi and PDO have their advantages:

PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.

So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.

Both are object-oriented, but MySQLi also offers a procedural API.

Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security.

**MySQL Examples in Both MySQLi and PDO Syntax**

In this, and in the following chapters we demonstrate three ways of working with PHP and MySQL:

* MySQLi (object-oriented)
* MySQLi (procedural)
* PDO