**The tutorial has three parts.**

* **Creating the example scripts from the document “Database Forms with PHP - Part 2”**
* **Answering the theory questions in part two (some of these will require additional research on the Internet).**
* **Developing the practical scripts in part three.**

**This tutorial includes practical and theoretical questions. It will require you to answer some questions based on existing knowledge, discussions with the teacher and research on selected web sites. The link below accesses the PDO section of the PHP manual.**

<http://php.net/manual/en/book.pdo.php>

**If the question is a written answer questions insert some space at the end of the question and include your answer in blue text. If the question requires a PHP script to be created, create the script and save it for later submission.**

**Part 1 – Practical Examples (exercises)**

Create each of the example programs in the document “**Database Forms with PHP - Part 2**”

**Part 2 - Theory Questions**

1. **What is the isset() function and how would we use it in our PHP programs?**

The isset() function determines if a variable is set and is not NULL. We often use it to determine if the submit button has been pressed.

1. **What is software reuse, and how would it be implemented in PHP programs?**

Software reuse means creating software applications by using predefined components. The predefined components could be code scripts, functions or even class objects. When we reuse these components, it will provide us with flexibility, facility and efficiency.

In PHP programs, we may first create some common classes with functions of database operation and then, allow the other classes in the same or different layers to access these common classes.

1. Are there any changes you can make to programs you created in this chapter to make them more reusable?
2. **What does the command “die” do?**

The die() function is similar to exit() function. It terminating the execution of current script and output a message.

**Part 3- Practical Question**

1. **Create a library file for validation functions and place it in the folder BLL. Modify the existing programs to move the validation code from the edit\_branch.php page into a function named validateBranch() in this library.**
2. **Modify the add\_branch.php page to add a call to the validateBranch() function to perform the same type of validation as the edit\_branch.php page**
3. **Modify the delete\_branch.php page or create a function in the validation library to check to make sure the program only allows deletion of Branches that do not have any associated vehicle records. You will need to query the m\_vehicle table to see if there are any vehicles for the Branch that will be deleted. If vehicles do exist, display a message to say the branch cannot be deleted.**