Lab 10 – Developing Object Oriented PHP

Reference: Chapter 9 of the "PHP Programming with MySQL" textbook.

PHP OOP http://php.net/manual/en/language.oop5.php

Aims:

- To be able to understand and develop code using Object Oriented Programming in PHP.

Getting Started:

Create a new folder '**lab10**' under the unit folder on the mercury server *~/cos30020/www/htdocs*.
Save today's work in this lab10 folder.

All Web pages must be validated.
You could also create and link an external stylesheet, to the pages, and this should be valid CSS3.

Task 1: Monster Class (9 marks)

The overall task is to create a simple PHP Monster class that allows one to create a monster with different number of eyes and colour. Then create a web application that demonstrates the use of the Monster class.

Step 1:
Create a file **`monsterclass.php`** that defines the monster class. This page does not contain any HTML tags and will have the following:

- Two properties namely number of eyes (integer) and colour (string)
- A constructor that initialises the number of eyes and colour.
- A set and get method for the number of eyes and colour (optional – not marked)
- A describe method that describes the monster

```
<?php
   ___(1)___ Monster {                  // start the Monster class
     ___(2)___ $num_of_eyes;             // properties
     ___(3)___ $colour;

   function ___(4)___($num, $col) {     // constructor
     ___(5)___;                         // initialise number of eyes
     ___(6)___;                         // initialise colour
   }

   function describe () {
     $ans = "The " . ___(7)___ . " monster has " . ___(8)___. " eyes.";
     return $ans;
   }
?>
```

Step 2:
Create a file **`monsterapp.php`** that uses the Monster class to create 2 monsters and describe them. The 2 monsters will have 1and 3 eyes with the colour "red" and "blue" respectively.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="description" content="Web application development" />
  <meta name="keywords" content="PHP" />
```

```
    <meta name="author"   content="Your Name" />
    <title>TITLE</title>
  </head>
  <body>
  <h1>Web Programming - Lab10</h1>
  <?php
      require_once ("monsterclass.php");       // include the monster class

      $monster1 = ___(9)___;           // creates a red monster with 1 eye
      $monster2 = ___(10)__;           // creates a blue monster with 3 eyes

      echo "<p>", ___(11)___,"</p>";  // describe the first monster
      echo "<p>", ___(12)___,"</p>";  // describe the second monster
  ?>
</body>
</html>
```

Task 2: Creating a simple "Hit Counter" (6 marks)

The overall task is to create a 'Hit Counter' that counts the number of 'hits' to a Web page and stores the results in a MySQL database. The functionality will be provided by creating a `HitCounter` class with a *data member* that stores the number of hits, and *set* and *get member functions* to access the counter member variables.

Step 1:
Create a PHP file **`setup.php`** that will

- store your MySQL connection details (host, username, password, databasename) and
- create the `hitcounter` database table.

The page contains a form with method **`post`** and action to **`setup.php`**, and with inputs for your username, password, databasename, and 'Set Up' submit.

Setup should then:

- process the 'posted' data to create on "feenix-mariadb.swin.edu.au" a "data/lab10" directory, and a text file (either "mykeys.txt", or "mykeys.inc.php") file, containing your connection details. *Hint: Adapt the files you created in Lab08 for this task.*
- create a "hitcounter" database table on "feenix-mariadb.swin.edu.au" with two fields `id` and `hits` (both small integers), and insert an initial value into the table.
    - CREATE TABLE `` `hitcounter` ( `id` SMALLINT NOT NULL PRIMARY KEY, `hits` SMALLINT NOT NULL ); ``
    - INSERT INTO `hitcounter` VALUES (1,0);
      Hint: Adapt the code from Lab08 for this task.
      It is OK for this script to use Procedural code.
- display a message when everything is setup OK.

Note:

- *If this task is too time consuming, copy your 'include' files, from previous Labs, into your lab10 folder, and*
- *Create the table 'hitcounter' using the MySQL Monitor, or phpMyAdmin*

Step 2:
Create the `HitCounter` class in PHP script `hitcounter.php` that will

- This class should include a **constructor** to initialize data members, and make a connection to the MySQL server, the database, and the table `hitcounter`.
- In other words, the constructor has parameters for the database connection: host, username, userpassword, dbname, tablename - then makes a connection to the database.
- Write the database connection in Object Oriented Syntax.
- Declare separate member functions:
  - `getHits()` – to get and display `hits` on the page,
  - `setHits()` – to add one to `hits`, and update the table,
  - `closeConnection()` – to close the connection
- Optionally also write a member function `startOver()` to set `hits` to zero.

Step 3:
Create the `countvisits.php` page that:

- 'require_once' the class code hitcounter.php,
- 'require_once' your connection details file mykeys.inc.php,
  OR code to read your mykeys.txt file.
- instantiates an object of the HitCounter class,
- uses this object to call the member functions of the HitCounter class,
- i.e. calls `getHits()` method,
  calls `setHits()`,method, and then
  calls the `closeConnection()` method.

Hit Counter

This page has received 7 hits.

For example, a `$Counter` object could call the `closeConnection( )` method of the `HitCounter` class to terminate the server connection, as follows:

```
$Counter->closeConnection();
```

Step 4:
Create the `startOver()` member function in the `HitCounter` class that sets `hits` back to zero

- Create the `startover.php` script that:
  - `'require_once'` the class code `hitcounter.php`,
  - `'require_once'` your connection details file `mykeys.php`,
    OR code to read your `mykeys.txt` file.
  - instantiates an object of the `HitCounter` class,
  - calls the `startOver()` method,
  - then calls the `closeConnection()` method.