LAB GOALS

- Interacting with MySQL
- PHP Classes and Objects (Constructors, Destructors, Instantiation, public, private, protected, ..)

Step 1: Start with creating a simple database using phpmyadmin. To do so, first make sure your development environment is running (e.g., WAMP, LAMP, XAMPP, XAMPP lite, etc). Then point your browser to http://localhost/phpmyadmin you should see the phpmyadmin interface:

Using the following SQL statement, create the following:

A. Using GUI or SQL, Create a new database called Lab17

```sql
CREATE DATABASE IF NOT EXISTS Lab17;
```

B. Create a new table named classes_taken, with 5 fields: (class, credits, semester, year, grade)

```sql
CREATE TABLE `lab17`.`classes_taken` (  
`Class` VARCHAR( 32 ) NOT NULL ,  
`Credits` INT NOT NULL ,  
`Semester` VARCHAR( 32 ) NOT NULL ,  
`Year` INT NOT NULL ,  
`Grade` VARCHAR( 5 ) NOT NULL  
) ENGINE = MYISAM ;
```

C. Using the GUI or SQL, insert three records in the classes_taken table.

```sql
INSERT INTO `classes_taken` (`Class`, `Credits`, `Semester`, `Year`, `Grade`) VALUES  
('ENG-W131', 3, 'Fall', 2011, 'A'),  
('INFO-I101', 4, 'Fall', 2011, 'B'),  
('SPCH-S121', 3, 'Spring', 2012, 'A');
```
<table>
<thead>
<tr>
<th><strong>D.</strong> Using SQL statements, create a new user account (&quot;Lab17User&quot;) for this lab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE USER 'Lab17User'@'localhost' IDENTIFIED BY 'Pass123Word';</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>E.</strong> The user will also get some privileges to define and manipulate the Lab17 database.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER ON <code>Lab17</code> . * TO 'Lab17User'@'localhost';</td>
</tr>
</tbody>
</table>
Step 2: Now that we have our database, let’s create a simple PHP class that manipulates the above database. We want a class with the following capabilities:

1) To be instantiated with the proper host, user, password and database credentials. (A constructor)
2) The ability to close the database when the object goes out of scope or when it is set to NULL. (A destructor)
3) To be able to add a record to the classes_taken table, by calling the method: Add_Class(Course, credit, semester, year, grade)
4) To be able to display all the classes taken up to this point by calling the method Print_Transcript()
5) To be able to display the total number of credits taken up to this point, by calling the method Total_Credits()

Let’s start with the structure of the class which accommodates the above:

```php
<?php
// Lab 17

class CoursesTaken {

    // Private Data
    private $HostName;
    private $UserID;
    private $Password;
    private $DBName;
    private $Con; // MySQL Connection

    // Constructor
    // Instantiating the object with proper host, user, password and database credentials
    public function __construct($host = NULL, $uid = NULL, $pw = NULL, $db = NULL) {
    }

    // Destructor
    // Close the connection
    public function __destruct() {
    }

    // To Add a record to the classes_taken table
    public function Add_Class($course, $credit, $semester, $year, $grade) {
    }

    // Display all the classes taken up to this point
    public function Print_Transcript() {
    }

    // Display the total number of credits taken up to this point
    public function Total_Credits() {
    }

    ?>

<?php
// Testing the class
$MyCourses = new CoursesTaken("localhost", "Lab17User", "Pass123Word", "Lab17");
?>
```

Run your program, at this point, you should see no output. Of course you should not see any errors either!
Step 2: Now all we have to do is to put some code in each of the above methods and test our class as we go:

Let’s start with our **Constructor** and **Destructor** methods.

```php
public function __construct($host = NULL, $uid = NULL, $pw = NULL, $db = NULL) {
    echo("The class constructor is being called... <br/>");
    $this->HostName = $host;
    $this->UserID   = $uid;
    $this->Password = $pw;
    $this->DBName   = $db;

    // Connect to Database
    $this->Con = mysqli_connect($host, $uid, $pw, $db);
    if (mysqli_connect_errno($this->Con)) {
        echo "Failed to connect to MySQL: ". mysqli_connect_error();
    } else {
        echo "Successful Connection to MySQL <br/>";
    }
}
```

```php
public function __destruct() {
    echo("The class destructor is being called... <br/>");
    // Close connection
    mysqli_close($this->Con);
}
```

```php
<?php
    // Testing the class
    $MyCourses = new CoursesTaken("localhost", "Lab17User", "Pass123Word", "Lab17");
?>
```

The class constructor is being called...
Successful Connection to MySQL
The class destructor is being called...
Step 3: Now, let's put in the code for `Print_Transcript()` method:

```php
// Display all the classes taken up to this point
public function Print_Transcript()
{
    $sql = "SELECT Class, Credits, Semester, Year, Grade
            FROM classes_taken
            ";

    $result = mysqli_query($this->Con,$sql);

    $arrayResult = array();

    echo("----------------Start Transcript----------------<br />");
    while($row = mysqli_fetch_array($result)) {
        $arrayResult[] = $row;  // Store the record to be returned
        echo($row['Class'] . " ".
             $row['Credits'] . " ".
             $row['Semester'] . " ".
             $row['Year'] . " ".
             $row['Grade']
        );
        echo "<br />
    }
    echo("----------------End Transcript----------------<br />
    return($arrayResult);
}
```

```php
// Testing the class

$MyCourses = new CoursesTaken("localhost", "Lab17User", "Pass123Word", "Lab17");
$MyCourses->Print_Transcript();

?>
```

The class constructor is being called...
Successful Connection to MySQL
--------------Start Transcript-------------
ENG-W131 3 Fall 2011 A
INFO-1101 4 Fall 2011 B
SPCH-S121 3 Spring 2012 A
--------------End Transcript--------------
The class destructor is being called...
Step 4: Now, let’s put in the code for Total_Credits() method:

```php
// Display the total number of credits taken up to this point
public function Total_Credits()
{
    $sql = "SELECT sum(Credits) AS Total_Credits
    FROM classes_taken
    ";

    $result = mysqli_query($this->Con,$sql);
    $arrayResult = array();
    echo("---------------- Start Total Credits ----------------<br />");
    while($row = mysqli_fetch_array($result)) {
        $arrayResult[] = $row;  // Store the record to be returned
        echo("Total credits completed = ", $row["Total_Credits"]);
    }
    echo("---------------- End Total Credits ----------------<br />");
    return($arrayResult);
}
```

```php
<?php
    // Testing the class
    $MyCourses = new CoursesTaken("localhost", "Lab17User", "Pass123Word", "Lab17");
    $MyCourses->Print_Transcript();
    $MyCourses->Total_Credits();
?>
```

The class constructor is being called...
The class destructor is being called...
Successful Connection to MySQL
---------Start Transcript---------
ENG-W131 3 Fall 2011 A
INFO-1101 4 Fall 2011 B
SPCH-S121 3 Spring 2012 A
---------End Transcript---------
---------Start Total Credits ---------
Total credits completed = 10
---------End Total Credits ---------
Step 5: Finally, let’s put in the code for Add_Class() method:

```php
// To Add a record to the classes_taken table
public function Add_Class($course, $credit, $semester, $year, $grade) {
    $sql = "INSERT INTO classes_taken (`Class`, `Credits`, `Semester`, `Year`, `Grade`)
    VALUES ('$course', $credit, '$semester', $year, '$grade');
    ",
    $result = mysqli_query($this->Con, $sql);
    if ($result == true) {
        echo "Successful Insert<br />
    } else {
        echo "Error Inserting class" . mysqli_error($this->Con) . "<br />
    }
}

<?php
    // Testing the class
    $MyCourses = new CoursesTaken("localhost", "Lab17User", "Pass123Word", "Lab17");
    $MyCourses->Print_Transcript();
    $MyCourses->Total_Credits();
    $MyCourses->Print_Transcript();
    $MyCourses->Total_Credits();
?>
```