Lab 3 – Functions and Control Structures

Reference: Chapter 2 of the "PHP Programming with MySQL" textbook.

PHP Control Structures: http://www.php.net/manual/en/language.control-structures.php
PHP User-defined Functions: http://www.php.net/manual/en/functions.user-defined.php

Aims:

- To be able to use various control structures and develop your own functions.

Getting Started:

Create a new folder '**lab03**' under the unit folder on the mercury server
~/cos30020/www/htdocs folder on mercury.
Save today's work in this lab03 folder.

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

Task 1: Using if and while statements (9 marks)

Step 1:
Create a file **`mathfunctions.php`** to contain a function called `factorial` that accepts a positive integer and returns its factorial value. A factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. For example,

5! = 5 x 4 x 3 x 2 x 1 = 120

```
<?php
  function factorial ($n) {   // declare the factorial function
      $result = 1;            // declare and initialise the result variable
      $factor = $n;           // declare and initialise the factor variable
      while ($factor > 1) {   // loop to multiple all factors until 1
        $result = $result * $factor;
        $factor--;            // next factor
      }                       // Note that the factor 1 is not multiplied
      return $result;
  }
?>
```

Step 2:
Create a file **`factorial.php`** that will include the file **`mathfunctions.php`** in order to access the defined functions in the file. It should also receive an input from **`factorialform.php`** from Step 3 via GET method, and check if the input is a positive integer then output its factorial value. Otherwise, it should generate an appropriate error message.

```
<!DOCTYPE html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="description" content="Web Application Development :: Lab 1" />
<meta name="keywords" content="Web,programming" />
<title>Using if and while statements</title>
</head>
<body>
<?php
  include ("mathfunctions.php");
?>
<h1>Web Programming - Lab 3</h1>
<?php
  if (isset ($_GET["________"])) {  // check if form data exists
    $num = $_GET["________"];       // obtain the form data
```

```
          if (____________________) {      // check if $num is a positive number
            if ($num == round ($num)) {    // check if $num is an integer
              echo "<p>", $num, "! is ", factorial ($num), ".</p>";
            } else {                       // number is not an integer
              echo "<p>Please enter an integer.</p>";
            }
          } else {                         // number is not positive
            echo "<p>Please enter a positive integer. </p>";
          }
        } else {                           // no input
          echo "<p>Please enter a positive integer.</p>";
        }
      ?>
      </body>

</html>
```

Step 3:
Create a file **`factorialform.php`** that contains a form with a single text box that allows a user to enter a number, and submit it to **`factorial.php`**.

```
<!DOCTYPE html>
<html lang="en" lang="en" >
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="description" content="Web Application Development :: Lab 3" />
<meta name="keywords" content="Web,programming" />
<title>Using if and while statements</title>
</head>
<body>
<h1>Web Application Development - Lab 3</h1>
<form action = _____________ method = _____________ >
   ____________________________________________
   ____________________________________________
   ____________________________________________
   ____________________________________________
   ____________________________________________
</form>
</body>
</html>
```

Test in the browser.

Task 2: Using `if` statement (3 marks)

Step 1:
Create a file **`leapyear.php`** with a script that tests if a variable value is a number, and if it is a leap year, and prints a message stating whether the year is a *standard year* or a *leap year*.

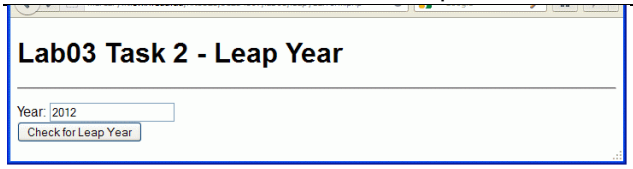
If the numerical value for a year is divisible by 4, it is a leap year. However, if the year is also divisible by 100 it is not a leap year, unless the year is also divisible by 400, in which case it is a leap year.

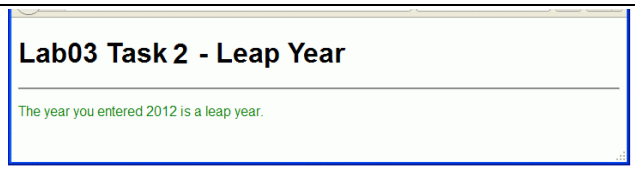
Test in the browser.

Step 2:
Create a file **`leapyearform.php`** that contains a form with a single text box that allows a user to enter a year, and submit it to **`leapyear.php`**.

Change **`leapyear.php`** to receive the year entered and determine if it is a leap year.

Test in the browser, and check that the pages are valid.





Step 3:
Modify the script in **`leapyear.php`** to contain a function **`is_leapyear`** that accepts a single parameter representing the year. The function returns true if the year is a leap year otherwise false.

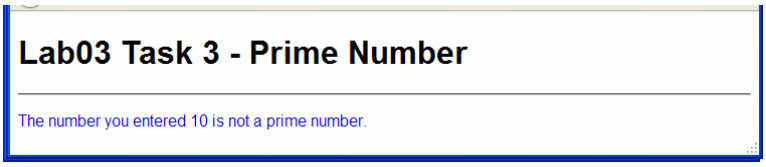
Test in the browser.

Task 3: Implementing loop statements (3 marks)

Step 1:
Create another file **`primenumber.php`** with a script that determines whether a number between 1 and 999, is a prime number and displays the result with an echo statement.

A prime number is a number that can only be divided by itself or by one. Examples of prime numbers include 1, 3, 5, 7, 13, and 17. You need to use a looping statement to test all division possibilities.

View in the browser.



Step 2:
Create a file **`primenumberform.php`** that contains a form with a single text box in which users can enter a number, and submit it to **`primenumber.php`**.

Change **`primenumber.php`** to receive the number entered and determine if it is a prime number.

Test in the browser.

Step 3:
Modify the script in **`primenumber.php`** to contain a function **`is_prime`** that accepts a single integer parameter and returns **true** it is a prime number otherwise **false**.

Test in the browser.

Extra Challenge:

Save a copy of **`leapyear.php`** as **`leapyear_selfcall.php`**.
Copy the form from **`leapyearform.php`** into **`leapyear_selfcall.php`** and change the form action to **`leapyear_selfcall.php`** and test.

To improve the user interface, check **`if`** no form input has been entered, using **`isset`** function, so that a check and display is only made when a value is submitted. http://php.net/manual/en/function.isset.php