University of Maryland College Park
Dept of Computer Science
CMSC389N Spring 2015
Midterm I Key

Last Name (PRINT): _____

First Name (PRINT): _____

University Directory ID (e.g., umcpturtle)_____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

Instructions

- This exam is a closed-book and closed-notes exam.
- Total point value is 200 points.
- The exam is a 50 minutes exam.
- Please use a pencil to complete the exam.
- WRITE NEATLY.
- **You don't need to use meaningful variable names; however, we expect good indentation.**

Grader Use Only

#1	Problem #1 (HTML/CSS/ PHP Language)	(60)	
#2	Problem #2 (PHP Coding)	(55)	
#3	Problem #3 (PHP Coding)	(85)	
Total	Total	(200)	

Problem #1, (HTML/CSS/PHP Language)

1. (3 pts) Using the <img> tag define a image entry where the image name is myPhoto.jpg and the message "Customer Photo" will appear when the image cannot be displayed.

   Answer: <img src="myPhoto.jpg" alt="Customer Photo">

2. (3 pts) Define a CSS rule that associates the color blue with links that have not been visited.

   Answer:

   ```
   a:visited {
         color: purple;
   }
   ```

3. (3 pts) What is the difference between an id selector and a class selector?

   Answer: An id selector applies to only one element; a class selector can be applied to several elements.

4. (3 pts) The following script is expected to display the area value. Are there any errors in the script? If there are no errors write NO_ERROR; otherwise, write ERROR and how to fix it.

   ```
   <?php
   $area = 200;
   task();
   function task() {
       echo "value is: $area";
   }
   ?>
   ```

5. (3 pts) Which of the following are considered false in PHP? Circle all that apply.

   a. 0
   b. "" (empty string)
   c. false
   d. 10
   e. "0"

   Answer: a., b., c., e.

6. (3 pts) What is the difference between == and === while comparing objects?

   Answer: = = true if two objects have same attributes and values and are instances of the same class; = = = true if and ony if the two variables refer to the same object instance

7. (3 pts) Using PHP define a constant called MAX_LENGTH that has a value of 50.

   Answer: define("MAX_LENGTH", 50);

8. (3 pts) What takes place when a file included using require_once is not found?

   Answer: A fatal error will take place and the script will be terminated.

9. (3 pts) What is an empty variable?

   Answer: a variable that has a value that evaluates to false.

10. (4 pts) Name two uses for the header() function discussed in lecture.

    Answer: Some examples:

    i. Redirection using "Location:"
    ii. Authentication

11. (4 pts) Name two differences that exits between **echo** and **print**.

    Answer: Some examples
    i. Print returns a value; echo does not
    ii. echo can print expressions separated by commas

12. (25 pts) Define a PHP class call **Phone_account** that represents a phone account. The specifications for the class are:

    a. The class has two instance variables called **$customer_name** and **$number.**
    b. A static variable named **$total_accounts** that keeps track of the number of objects created.
    c. A constructor that initializes a **Phone_account** object. It has a name and phone number as parameters.
    d. A toString method that prints the name and number associated with a customer object. See the output below for format information.

    Answer:

```
class Phone_account {
    private $customer_name, $number;
    static $total_accounts = 0;

    public function __construct($customer_name, $number) {
        $this->customer_name = $customer_name;
        $this->number = $number;
        Phone_account::$total_accounts++;
    }

    public function __toString() {
        return "Name: ".$this->customer_name.", Number: ".$this->number;
    }
}
```

Problem #2, (PHP Coding)

Write a PHP function called **generate_list** that has the following specifications:

1. Three parameters
    a. $data → Associative array that maps strings to strings
    b. $print_keys → Boolean value
    c. $header → string
2. The function will generate HTML representing an **ordered** HTML list. If the $print_keys parameter has a true value, the function will use the keys of the $data array; otherwise the values of the $data array will be used.
3. The $header represents a header (displayed using <h1></h1>) that will appear before the list.
4. The default value for the $print_keys parameter is true.
5. The default value for the $header parameter is "Generic List".
6. If the $data array is empty, the function will just return a header (display using <h1></h1>) with the message "Empty List".
7. The following is an example of calling the function you will write. This is just an example and your function should work for other arrays.

Answer:

```
function generate_list($data, $print_keys = true, $header = "Generic List") {
	if (!isset($data)) {
		return "<h1>Empty List</h1>";
	} else {
		$answer ="<h1>".$header."</h1><ol>";
		$elements = $print_keys ? array_keys($data) : array_values($data);
		foreach ($elements as $entry) {
			$answer .= "<li>$entry</li>";
		}
		$answer .= "</ol>";
		return $answer;
	}
}
```

Problem #3, (PHP Coding)

Write a PHP script that generates a form that computes the square of values provided and the number of times the script is executed. For this problem:

1. Define a header called "Square Calculator" using <h1></h1>.
2. Define a text field that allow us to enter a number. The text "Value: "should appear to the left of the text field. The default value for the text field will be 5.
3. A submit button named "Square" will allow us to trigger the computation of the square. The result will be displayed in the text field. For example, if the current value in the text field is 5 and we press the button, 5 will be replaced with 25.
4. The number of times the script has been executed will be printed after the "Square" button.
5. The name of the script is compute.php.
6. Your script must be a self-referencing script. You may not add any other script file.
7. You must use a heredoc in order to generate the form.
8. Use the post method to submit your form.
9. You must use a hidden field to keep track of the number of times the script is executed.
10. The "support.php" file has the **generatePage** function that takes the body of an HTML document and generates a complete document. This is the same function we presented in class. Use it to generate the final document that will be displayed. For example, if $body has the HTML body, you will call the function as follows: ***echo generatePage($body);***
11. An example of the form is provided below.

Square Calculator

Value: 5

Square

Times Executed: 0

Answer:

```
<?php
        require_once("support.php");

        if (isset($_POST["Compute"])) {
                $answer = ($_POST["value"] * $_POST["value"]);
                $times_executed = $_POST["times_executed"] + 1;
        } else {
                $answer = "5";
                $times_executed = "0";
        }

        $body = <<<EOBODY
            <h1>Square Calculator</h1>
                <form action="compute.php" method="post">
                    <strong>Value: </strong><input type="text" name="value" value="$answer"><br><br>
                        <input type="submit" name="Compute" value="Square">
                        <input type="hidden" name="times_executed" value="$times_executed" />
                </form>
                <br>Times Executed: $times_executed
EOBODY;

        echo generatePage($body);
?>
```