



University of Maryland College Park

Dept of Computer Science

CMSC389N Summer 2016

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 tag define a image entry where the image name is city.jpg and the message "College Park" will appear when the image cannot be displayed.

Answer:

2. (3 pts) Define a CSS rule that associates the color yellow with links that have been visited.

Answer:

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

3. (3 pts) Define a CSS rule that associates with paragraphs the color yellow and the size 3 em.

```
p {
    font-size: 3em;
    color: yellow;
}
```

4. (3 pts) What is HTTP?

Answer: Protocol that defines how user agents (e.g., browser) and web server can communicate. Accept as valid Hypertext Transfer Protocol

5. (3 pts) Provide an example of what is possible by using server side includes.

Some choices: file inclusion, timestamp for file modification, timestamp for local date / time or any other valid one they may have provided.

6. (3 pts) In HTML what is the difference between an inline element and a block element?

Answer: Block elements begin on new lines whereas inline elements don't. Any other valid difference receives full credit.

7. (3 pts) Name one difference between PHP's echo and print.

Answer: Some differences

- i. Print returns a value; echo does not
- ii. echo can print expressions separated by commas

8. (3 pts) Which of the following are VALID variable names (will not cause an error during script execution) in PHP? Circle the valid ones.

- a. \$age_55
- b. \$LOCATION
- c. Area
- d. window_position\$

Answer: a. and b.

9. (3 pts) Using PHP define a constant called MAX_LENGTH that has a value of 200.

Answer: `define("MAX_LENGTH", 200);`

10. (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.

11. (3 pts) Name three PHP superglobals seen in class.

Answer: Any three of `$_GET`, `$_POST`, `$_COOKIE`, `$_FILES`, `$_SERVER`, `$_ENV`, `$_REQUEST`, `$_SESSION`

12. (3 pts) We have defined a PHP function as follows:

```
function process() { echo "hello"; }
```

Which of the following are VALID ways to call the function?

- a. `Process();`
- b. `process();`
- c. `PROcess();`
- d. None of the above.

Answer: a., b., c.

13. (24 pts) Define a PHP class call **App** that represents a web app. The specifications for the class are:
- The class has two instance variables called **\$name** and **\$id**.
 - A static variable named **\$downloaded** that keeps track of the number of times the app has been downloaded.
 - A constructor that initializes an **App** object. It has a name and id number as parameters.
 - A toString method that prints the name and id associated with an App object. See the output below for format information.
 - A download() method that increases the number of downloaded apps.
 - A getDownloaded method that returns the number of downloaded apps.

Answer:

```
class App {
    private $name, $_id;
    private static $downloaded = 0;

    public function __construct($name, $id) {
        $this->name = $name;
        $this->id = $id;
    }

    public function download() {
        App::$downloaded++;
    }

    public function getDownloaded() {
        return App::$downloaded;
    }

    public function __toString() {
        return "Name: ".$this->name.", Id: ".$this->id;
    }
}
```

Problem #2, (PHP Coding)

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

- Two parameters
 - \$data → Associative array that maps strings to integers.
 - \$header → string
- The function will generate HTML representing an HTML table.
- The \$header represents a header (displayed using <h1></h1>) that will appear before the table.
- The default value for the \$header parameter is “Generic Table”.
- If the \$data array has not been set, the function will just return a header (displayed using <h1></h1>) with the message “Empty Table”.
- 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 and header values.

Answer:

```
function generate_table($data, $header = "Generic Table") {
    if (!isset($data)) {
        return "<h1>Empty Table</h1>";
    } else {
        $keys = array_keys($data);
        $answer = "<h1>". $header. "</h1>";
        $answer .= "<table border='1'>";
        foreach ($keys as $entry) {
            $answer .= "<tr>";
            $answer .= "<td>". $entry. "</td><td>$. $data{$entry} .</td>";
            $answer .= "</tr>";
        }
        $answer .= "</table>";

        return $answer;
    }
}
```

Problem #3, (PHP Coding)

Write a PHP script that generates a form that computes the sum or product of values provided via a form. For this problem:

1. Define a header called “Calculator” using `<h1></h1>`.
2. Define two text fields that allow us to enter two numbers. The text “Value1: “ and “Value2:” should appear to the left of each text field. The default values for each text field will be 2 and 3, respectively.
3. Two submit buttons named “sum” and “product” will allow us to trigger the appropriate computation. The result will be displayed below the buttons after the message “Result:”.
4. When the form is initially processed the value after “Result:” will be “None”.
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 `get` method to submit your form.
9. 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 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);`
10. An example of the form is provided below.

Before Selecting Any Buttons

Calculator

Value1: Value2:

Result: None

After Pressing sum Button (not showing when we select product)

Calculator

Value1: Value2:

Result: 5

WRITE YOUR SCRIPT ON THE NEXT PAGE

Answer:

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

    $answer = "None";
    if (isset($_GET["sum"])) {
        $answer = ($_GET["value1"] + $_GET["value2"]);
    } elseif (isset($_GET["product"])){
        $answer = ($_GET["value1"] * $_GET["value2"]);
    } else {
        $answer = "None";
    }

    $body = <<<EOBODY
        <h1>Calculator</h1>
        <form action="compute.php" method="get">
            Value1: <input type="text" name="value1" value="2">
            Value2: <input type="text" name="value2" value="3"><br><br>
            <input type="submit" name="sum" value="sum">
            <input type="submit" name="product" value="product">
        </form>
        <br>Result: $answer
    EOBODY;

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