



University of Maryland College Park

Dept of Computer Science

CMSC389N Spring 2017

Midterm I Solutions

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 75 minutes exam.
- Please use a pencil to complete the exam.
- WRITE NEATLY.
- **You may not use any JavaScript nor Bootstrap.**
- **You don't need to use meaningful variable names; however, we expect good indentation.**

Grader Use Only

| | | | |
|--------------|-----------------------------|-------|--|
| #1 | Problem #1 (Miscellaneous) | (30) | |
| #2 | Problem #2 (PHP Coding I) | (35) | |
| #3 | Problem #3 (PHP Coding II) | (50) | |
| #4 | Problem #4 (PHP Coding III) | (85) | |
| Total | Total | (200) | |

Problem #1 (HTML/CSS Concepts)

1. (3 pts) Name one difference between an id selector and a class selector.

Answer: Class selector allow us to apply the same rules to many elements; an id selector allows to apply rules to only one element. Another difference: class selectors are defined using #; id selectors using a period.

2. (3 pts) Name two uses we saw in class for the header function.

Answer: Any of the following:

Redirection to a site (e.g., `header("Location:http://www.cs.umd.edu")`)

Authentication (to ask for a username and password via a pop-up window)

3. (3 pts) When is a variable considered an empty variable?

Answer: When it has a value that evaluates to false.

4. (3 pts) Which of the following unsets a variable? Circle all that apply.

- a. Assigning null to the variable.
- b. Assigning 0 to the variable.
- c. Using unset on the variable.
- d. Assigning the empty string to the variable.

Answer: a. and c.

5. (3 pts) What is one common use for server side includes?

Answer: Any of the following:

File inclusion

Timestamp for file modification

Timestamp for local date/time

6. (3 pts) What is the difference between == and ===? Briefly explain.

Answer: Two possible answers:

- a. === provides equivalency with regards to both value and type (e.g., "20" === 20 is false).
- b. == true if two objects have same attributes and values and are instances of the same class; === true if and only if the two variables refer to the same object instance.

7. (3 pts) Name one reason why you want to use post over get when submitting data to a server.

Answer: Any of:

Security (password information will not be in the URL)

Amount of information to be sent is large

The request changes the state of the server (e.g., adding or removing data from a database)

8. (3 pts) Write a SQL command that will create a table named "scores" that has the fields studentsName (string 30 characters), score (integer) and maxScore (integer).

Answer: `create table scores (studentsName varchar(30), score integer, maxScore integer);`

9. (3 pts) Write a SQL command that will insert a record into the “scores” table above for a student whose name is “Kyle”, with a score of 80, in a test with a maximum score of 200.

Answer: insert into scores values (“Kyle”, 80, 200);

10. (3 pts) Write a SQL command that will display the name and score (not maxScore) of students where the maximum score of the test is 200.

Answer: select studentsName, score from scores where maxScore = 200;

Problem #2 (PHP Coding)

Define two PHP classes called **Lamp** and **ElectricLamp**. The specification for these classes are:

1. **Lamp** class
 - a. The class has two instance variables called **\$brand** and **\$size**.
 - b. A static variable named **\$totalLamps** keeps track of the number of Lamp objects created.
 - c. A constructor that initializes a **Lamp** object. It has a brand and size number as parameters.
 - d. A toString method that prints the brand and size associated with a **Lamp** object. See the output below for format information.
 - e. getBrand – get method for brand field.
 - f. getTotalLamps – static method that returns total number of **Lamp** objects created.
 - g. Use strict types for parameters and return values. You can assume the declaration **declare(strict_types=1)** already exists.

Answer:

```
class Lamp {
    private $brand;
    private $size;
    static $totalLamps = 0;

    public function __construct(string $brand, int $size) {
        $this->brand = $brand;
        $this->size = $size;
        Lamp::$totalLamps++;
    }

    public function __toString() : string {
        return "Brand: \"\".\"$this->brand.\"\", Size: \"\".\"$this->size;
    }

    public function getBrand() : string {
        return $this->brand;
    }

    public static function getTotalLamps() : int {
        return Lamp::$totalLamps;
    }
}
```

2. **ElectricLamp** class

- The class extends the **Lamp** class and has an instance variable called **\$watts**.
- A constructor that has brand, size, and watts as parameters. The constructor will call the superclass constructor in order to initialize the brand and size.
- A toString method that prints the brand, size, and watts associated with the **ElectricLamp** object. See the output below for format information.
- Use strict types for parameters and return values. You can assume the declaration **declare(strict_types=1)** already exists.

Answer:

```
class ElectricLamp extends Lamp {
    private $watts;

    public function __construct(string $brand, int $size, int $watts) {
        parent::__construct($brand, $size);
        $this->watts = $watts;
    }

    public function __toString() : string {
        $lampStr = parent::__toString();
        return $lampStr.", Watts: ".$this->watts;
    }
}
```

Problem #3 (PHP Coding)

Write a PHP script that generates a form that computes a table with the squares of values in a range. For this problem:

1. Define two text fields that allow us to enter two numbers. The text “LowerLimit: “ and “UpperLimit:” should appear to the left of each text field. The default values for each text field will be 1 and 5, respectively.
2. Two submit buttons named “Print table” and “Clear” will allow us to generate a table of squares or to clear/remove the table.
3. The name of the script is print_table.php.
4. Your script must be a self-referencing script. You may not add any other script files.
5. You must use a heredoc in order to generate the form.
6. Use the get method to submit your form.
7. The text of odd-numbered rows should be in red. Remember the style attribute allows you to define CSS rules.
8. You can use a border size of 1 for the table.
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 executing the script is provided below.

Before Selecting Any Buttons

LowerLimit: UpperLimit:

After Pressing the *Print table* Button

LowerLimit: UpperLimit:

| Value | Square |
|-------|--------|
| 1 | 1 |
| 2 | 4 |
| 3 | 9 |
| 4 | 16 |
| 5 | 25 |

After Pressing the *Clear* Button

LowerLimit: UpperLimit:

Answer:

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

    $answer = "";
    if (isset($_GET["print_table"])) {
        $answer = "<table border='1'";
        $answer .= "<tr><th>Value</th><th>Square</th></tr>";

        for ($i = $_GET["value1"]; $i <= $_GET["value2"]; $i++) {
            if ($i % 2 == 0) {
                $answer .= "<tr>";
            } else {
                $answer .= "<tr style='color:red'>";
            }
            $answer .= "<td>$i</td><td>". $i * $i . "</td>";
            $answer .= "</tr>";
        }
        $answer .= "</table>";
    } if (isset($_GET["clear_table"])) {
        $answer = "";
    }

    $body = <<<EOBODY
        <form action="print_table.php" method="get">
            LowerLimit: <input type="text" name="value1" value="1">
            UpperLimit: <input type="text" name="value2" value="5"><br><br>
            <input type="submit" name="print_table" value="Print table">
            <input type="submit" name="clear_table" value="Clear">
        </form>
        <br>
        $answer
    EOBODY;

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

Problem #4 (PHP Coding)

For this problem you will define two PHP scripts that support the processing of data that is present a file. Each line of the file has the name of a person that has received consulting services from a company. The specification for this problem are:

1. Define a script called **process_requests.php** that will open a file specified via a text field. The script will read each line from the file and send the following information to a second script called **report.php**:
 - a. The names of individuals that need to pay for services. Only one person (John_Smith) does not pay for services and his name will not be sent to the **report.php** script. Notice that any name can appear multiple times in the file (including John_Smith) and in the data sent to report.php.
 - b. The total number of free requests (those associated with John_Smith) found.
 - c. The total number of paid requests found.
 - d. The default value for the text field will be service_requests.txt.
2. You must use sessions in order to send data from **process_requests.php** to **report.php**. If you don't, you will lose significant credit.
3. If the file cannot be opened the script will end with the message "Could not open file". Notice that spaces could surround names in the file therefore you should trim the data read.
4. Define a script called **report.php** that will retrieve the information sent by **process_requests.php** (using sessions) and will display the following report:
 - a. Free Requests: <NUM_FREE_REQS> where <NUM_FREE_REQS> represents those associated with John_Smith.
 - b. Paid Requests: <NUM_PAID_REQS> where <NUM_PAID_REQS> represents paid requests.
 - c. For each person (except John_Smith) display the name, followed by the number of requests (in parentheses), followed by the total amount of money due for services rendered. The following fees (per request) will be used:
 - i. Mary_Robinson:\$50
 - ii. Peter_Jackson: \$40
 - iii. Any person other than Mary_Robinson or Peter_Jackson: \$100
 - d. Notice that information about each person can appear in the report in any order.
5. You will lose significant credit if the second scripts opens and reads data from the file.

For this problem feel free to use the generatePage() function we saw in class that allows you to generate an HTML document when you provide the body (e.g., generatePage(\$body)). Assume this function is in the file support.php (make sure you include it). The following is an example of running the scripts assuming a file called **service_requests.txt** with the following entries:

```
John_Smith
Mary_Robinson
Peter_Jackson
Mary_Robinson
John_Smith
Peter_Jackson
Peter_Jackson
Charles_Sanders
```

The following is the form displayed by **process_requests.php**.

Filename:

The following is the output generated by **report.php** after pressing submit and execution of the **report.php** has been completed.

Free Requests: 2
Paid Requests: 6
Mary_Robinson: (2) \$100
Peter_Jackson: (3) \$120
Other: (1) \$100

process_requests.php

Answer:

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

    if (isset($_POST["submitInfoButton"])) {
        $fp = fopen($_POST["filename"], "r") or die("Could not open file");
        session_start();
        $_SESSION['names'] = array();
        $to_process = $ignored = 0;
        $line = trim(fgets($fp));
        while (!feof($fp)) {
            if ($line != "John_Smith") {
                $_SESSION['names'][] = $line;
                $to_process++;
            } else {
                $ignored++;
            }
            $line = trim(fgets($fp));
        }
        fclose($fp);
        $_SESSION['to_process'] = $to_process;
        $_SESSION['ignored'] = $ignored;
        header("Location: report.php");
    } else {
        $body = <<<EOBODY
            <form action="$_SERVER[PHP_SELF]" method="post">
                <strong>Filename: </strong><input type="text"
name="filename" value="service_requests.txt" />
                <input type="submit" name="submitInfoButton" />
            </form>
        EOBODY;
    }

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


report.php

Answer:

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

session_start();
$mary = $peter = $other = 0;
foreach ($_SESSION['names'] as $entry) {
    if ($entry == "Mary_Robinson") {
        $mary++;
    } else if ($entry == "Peter_Jackson") {
        $peter++;
    } else {
        $other++;
    }
}
$body = "<h1>Report</h1>";
$body = "Free Requests: " . $_SESSION['ignored'] . "<br>";
$body .= "Paid Requests: " . $_SESSION['to_process'] . "<br>";
$body .= "Mary_Robinson: ($mary) $" . ($mary * 50) . "<br>";
$body .= "Peter_Jackson: ($peter) $" . ($peter * 40) . "<br>";
$body .= "Other: ($other) $" . ($other * 100) . "<br>";

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