



University of Maryland College Park

Dept of Computer Science

CMSC389N Summer 2016

Midterm II 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 75 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/JS Language)	(40)	
#2	Problem #2 (PHP Coding)	(70)	
#3	Problem #3 (JavaScript Coding)	(30)	
#4	Problem #4 (JavaScript Coding)	(60)	
Total	Total	(200)	

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

1. (3 pts) In PHP what is the name associated with methods that start with two underscores?

Answer: Magic methods

2. (3 pts) Which of the following applies to HTTP?

- a. Stateless protocol
- b. Relies on HTML
- c. It is secure by default
- d. None of the above

Answer: a.

3. (3 pts) Session information is stored:

- a. In files in the server (one file per session)
- b. In files in the browser
- c. In a database
- d. None of the above

Answer: a.

4. (3 pts) Provide an alternative on how to have sessions when cookies are disabled in a browser.

Answer: Using hidden fields or using the URL to pass the session id.

Grading: Any reasonable answer receives full credit.

5. (3 pts) In JavaScript which value is associated with object properties that do not exist?

- a. undefined
- b. null
- c. a. and b.
- d. None of the above.

Answer: a.

6. (3 pts) Which of the following expressions are true in JavaScript?

- a. NaN == NaN
- b. NaN === NaN
- c. "20" == 20
- d. None of the above.

Answer: c.

7. (10 pts) Write a SQL command that will create a table named “friends” that has the fields name (string 10 characters), age (integer) and salary (float).

Answer: create table friends (name varchar(10), age int, salary float);

8. (6 pts) Write a SQL command that will insert a record in the “friends” table above for a friend named “Bob” that is 30 years old and that has a salary of \$50000.

Answer: insert into friends values ("Bob", 30, 50000);

9. (6 pts) Write a SQL command that will display the name and age of friends that have a salary greater than or equal to 20000.

Answer: select name, age from friends where salary >= 20000;

Problem #2 (PHP Coding)

Define two php scripts that will allow the processing of data present in a text file associated with an app store. Each line of the text file has one of the following words: map, game or educ. Each line represents an app that was bought from the store. For this problem:

1. Define a script called **readData.php** that will open the file specified via a text field. The script will read each line from the file and send that information (using sessions) to a second script called **process.php**. The script relies on the post method and it is a self-referencing script.
2. Define a script called **process.php** that will retrieve the lines (using sessions) and will display a summary of how many apps of each kind (map, game and educ) were sold. The second script MAY NOT open and read 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). **You may not use JavaScript for this problem. If you do not use sessions you will lose significant credit.**

The following is an example of running the scripts assuming a file called data.txt with the following entries:

map
map
educ
game
game
educ
educ
educ

The following is the form displayed by readData.php after we have entered “data.txt” in the text field.

Filename:

After pressing submit and execution of the process.php has been completed.

Summary

map: 2
game: 2
educ: 4

1. readData.php

Answer:

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

    if (isset($_POST["submitInfoButton"])) {
        session_start();
        if (!file_exists($_POST["filename"])) {
            $body = "<strong>File {$_POST['filename']} does not exist.</strong>";
        } else {
            $body = "<h1>Displaying file contents of file {$_POST['filename']}</h1>";
            $fp = fopen($_POST["filename"], "r") or die("Could not open file");
            $_SESSION['names'] = array();
            while (!feof($fp)) {
                $_SESSION['names'][] = trim(fgets($fp));
            }
            fclose($fp);
            header("Location: process.php");
        }
    } else {
        // superglobals are not accessible in heredoc
        $scriptName = $_SERVER["PHP_SELF"];
        $body = <<<EOBODY
            <form action="$scriptName" method="post">
                <p>
                    <strong>Filename: </strong><input type="text" name="filename" />
                </p>

                <!--We need the submit button-->
                <p>
                    <input type="submit" name="submitInfoButton" />
                </p>
            </form>
        EOBODY;

    }

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

2. process.php

Answer:

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

    session_start();
    $map = $game = $educ = 0;
    foreach ($_SESSION['names'] as $entry) {
        if ($entry == "map") {
            $map++;
        } else if ($entry == "game") {
            $game++;
        } else {
            $educ++;
        }
    }
    $body = "<h1>Summary</h1>";
    $body .= "map: " . $map . "<br>";
    $body .= "game: " . $game . "<br>";
    $body .= "educ: " . $educ . "<br>";

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

Problem #3 (JavaScript Coding)

Write **JavaScript (NOT PHP)** that defines two “classes” using the approach presented in class. **If you use E6 class definitions (similar to what you have in Java) you will not receive any credit for this problem.**

1. Bed

- Define a Bed “class” that has two **private** variables named brand and size.
- Define a constructor that has two parameters, brand and size.
- Define two get methods, getBrand and getSize, that return the brand and size, respectively.
- The Bed class has a prototype that defines a country variable with the value “USA” and an info() method that prints the brand and size (see example below for format information).

2. WaterBed

- Define a WaterBed “class” that “extends” the Bed class. The class has a **private** variable named waterCapacity.
- Define a constructor that has three parameters: brand, size and waterCapacity.
- Define a get method named getWaterCapacity that returns the water capacity.
- You need to define the appropriate prototype for this “class”.

The following is an example of using the “classes” you will define.

```
var bed1 = new Bed("Nealy", "King");

/* Regular Bed */
document.writeln("Info<br>");
bed1.info();
document.writeln("End Info<br><br>");
document.writeln("Bed Brand: " + bed1.getBrand() + "<br>");
document.writeln("Bed Size: " + bed1.getSize() + "<br>");
document.writeln("Bed Country: " + bed1.country + "<br><br>");

/* Water Bed */
```

```

waterBed = new WaterBed("Wealy", "Twin", 40);
document.writeln("WaterBed Brand: " + waterBed.getBrand() + "<br>");
document.writeln("WaterBed Size: " + waterBed.getSize() + "<br>");
document.writeln("WaterBed Country: " + waterBed.country + "<br>");
document.writeln("WaterBed WaterCapacity: " + waterBed.getWaterCapacity() + "<br>");
waterBed.info();

```

Output

```

Info
Brand: Nealy, Size: King
End Info

Bed Brand: Nealy
Bed Size: King
Bed Country: USA

WaterBed Brand: Wealy
WaterBed Size: Twin
WaterBed Country: USA
WaterBed WaterCapacity: 40
Brand: Wealy, Size: Twin

```

Answer:

```

function Bed(brand, size) {
    this.getBrand = function() {
        return brand;
    }
    this.getSize = function() {
        return size;
    }
}

Bed.prototype = {
    constructor: Bed,
    country: "USA",
    info: function() {
        document.write("Brand: " + this.getBrand());
        document.write(", Size: " + this.getSize() + "<br>");
    }
};

function WaterBed(brand, size, waterCapacity) {
    /* Calls super class constructor */
    Bed.call(this, brand, size);

    this.getWaterCapacity = function() {
        return waterCapacity;
    }
}

WaterBed.prototype = Object.create(Bed.prototype);

```

Problem #4 (JavaScript Coding)

Write a **JavaScript (NOT PHP)** program that allow us to display images (jpg files) in a table. Each row will display two images. For this problem:

- Define a form with a text field and a button (see example below for format information).
- The text field has a default value of 4.
- If the user enters an odd number the message “Invalid value” will be display below the “Images” header. Otherwise a table with the images (two per row) will be displayed below the “Images” header. You can assume the user will provide a number that will not exceed the number of images available. All images have **college** as base name, therefore images are named college1.jpg, college2.jpg, etc. The program and the images are in the same folder.
- A **main** function will define a function named **displayTable** as the function the **displayImages** button will call when selected.
- You can assume the user will enter a number greater than or equal to 1.
- Notice that the HTML and JavaScript appears in a single file.
- You can assume the width and height of each image is 100 pixels.
- Feel free to add any functions in addition to the displayTable function. For example, a function that returns a string with the tag initialized with the appropriate data can simplify your implementation. Notice you do not need to add the alt attribute to the tag.

Form

Viewer

Number of Images:

Images

After Providing Invalid Number and Clicking on the Button

Viewer

Number of Images:

Images

Invalid value

After Providing a Valid Number and Clicking on the Button

Viewer

Number of Images:

Images



Answer:

```
<body>
  <h2>Viewer</h2>
  Number of Images: <input id="limit" type="text" value=4>
  <input type="submit" id="process" value="displayImages"><br><br>
  <h2>Images</h2>
  <div id="display"></div>
  <script>
    "use strict";

    main();

    function genImage(basename, number) {
      var name = basename + "" + number + ".jpg";
      return "<img src=\"\" + name + \"\" width=\"100\" height=\"100\"/>";
    }

    function main() {
      var buttonInHTMLForm = document.getElementById("process")
      buttonInHTMLForm.onclick = displayTable; // DO NOT PUT ()
    }

    function displayTable() {
      var value = document.getElementById("limit").value, i;
      var body;

      if (value % 2 != 0 || value === "") {
        body = "<strong>Invalid value</strong>";
      } else {
        body = "<table border='1'>";
        for (i = 1; i <= value; i += 2) {
          body += "<tr>";
          body += "<td>" + genImage("college", i) + "</td>" + "<td>"
            + genImage("college", i + 1) + "</td>";
          body += "</tr>";
        }
        body += "</table>";
      }
      document.getElementById("display").innerHTML = body;
    }
  </script>
</body>
```