

Portion of test	Points possible	Points missed	Points correct
Written:	60		
Code Modification:	20		
Debug/Coding:	20		
Total:	100		

Student's Name: _____

East Tennessee State University – Department of Computer and Information Sciences
 CSCI 2910 (Tarnoff) – Server/Client Side Programming
 TEST 3 for Spring Semester, 2007

Read this before starting!

- The total possible score for this test is 100 points.
- This test is ***closed book and closed notes***. A summary sheet is attached to the back of this test. You may remove it from the staple to use as a reference while taking this test.
- The test consists of two parts: a 50 point written portion (this document) and a 50 point computer-based portion. You must complete the written portion of the exam before you are able to begin the computer-based portion.
- For the written portion, all answers must be placed in space provided. Failure to do so will result in no credit for answer.
- If you perform written work on the back of a page in this test, indicate that you have done so in case the need arises for partial credit to be determined.
- For the computer-based portion, ***you are allowed to use only the following applications on your lab PC:***
 - ***Either Dreamweaver or Notepad for editing your web pages***
 - ***A browser of your choice for viewing the web pages you've created and accessing the class web site and www.php.net/manual as a reference. No other searching or browsing should be necessary and therefore is prohibited.***
- Statement regarding academic misconduct from Section 5.7 of the East Tennessee State University Faculty Handbook, June 1, 2001:

"Academic misconduct will be subject to disciplinary action. Any act of dishonesty in academic work constitutes academic misconduct. This includes plagiarizing, the changing or falsifying of any academic documents or materials, cheating, and the giving or receiving of unauthorized aid in tests, examinations, or other assigned school work. Penalties for academic misconduct will vary with the seriousness of the offense and may include, but are not limited to: a grade of 'F' on the work in question, a grade of 'F' of the course, reprimand, probation, suspension, and expulsion. For a second academic offense the penalty is permanent expulsion."

All problems are worth 2 points each unless otherwise stated.

1. For each of the following needs or characteristics of a web application, identify which would be better suited, a server-side application or a client-side application. (1 point each)

Restricting a client's access to source code:	<input checked="" type="checkbox"/> Server-side	<input type="checkbox"/> Client-side
Eliminating browser dependency:	<input checked="" type="checkbox"/> Server-side	<input type="checkbox"/> Client-side
Quick validation of form data:	<input type="checkbox"/> Server-side	<input checked="" type="checkbox"/> Client-side
Smaller files to download:	<input checked="" type="checkbox"/> Server-side	<input type="checkbox"/> Client-side
Interactive web page content:	<input type="checkbox"/> Server-side	<input checked="" type="checkbox"/> Client-side
Better support for complex projects:	<input checked="" type="checkbox"/> Server-side	<input type="checkbox"/> Client-side
Independence from client's operating system:	<input checked="" type="checkbox"/> Server-side	<input type="checkbox"/> Client-side

2. True or **False:** PHP ignores whitespace (carriage returns, tabs, etc.) between quotation marks when defining a string.

PHP does not collapse white space into a single space like HTML and JavaScript do. Instead, it keeps all tabs, carriage returns, multiple spaces, etc. contained within the string. It seems a little deceptive because when you send the PHP output to an HTML browser, the browser collapses all of the white space. The PHP, however, does not.

3. True or **False:** PHP code contained between the tags `<?php` and `?>` is processed in files with the ".html" extensions when they are downloaded from the server .

The PHP extension alerts the PHP engine to process the scripts between the `<?php` and `?>` tags. Other extensions are not processed.

4. Which output function, *print* or **echo**, allows for multiple arguments separated by commas?
5. What is the output that results from the following set of PHP statements?

```
$s = "Thomas";  
print "My name is ".$s;
```

My name is Thomas

6. What is the output that results from the following set of PHP statements?

```
$p = 78.25;  
print "The item's value is \$$p.";
```

The item's value is \$78.25.

Note that the `\$` represents an escape character meant to output a single '\$'.

7. What is the output that results from the following set of PHP statements?

```
$a = array(5, 3, 4);  
print "The value of the array is $a.";
```

The value of the array is Array.

The function print does not output the array properly. Instead, it simply outputs the variable's type, i.e., Array.

8. Of the following characters, **circle all** of the ones that must be escaped in a PHP string that is defined using double quotation marks, i.e., "...". (3 points)

a.) - b.) & c.) " d.) \ e.) . f.) \$ g.) ? h.) *

9. True or False: Within a string defined in PHP, single quotes can be used without escaping within double quoted strings and vice versa.

10. **Circle all** of the **valid** PHP variable names from the list below. (3 points)

a.) \$5b b.) salary c.) _count d.) \$_i e.) \$i f.) \$var1

11. What is the output that results from the following set of PHP statements?

```
$var1 = 46.94;  
$var2 = (int)$var1;  
print "The result is ".$var2;
```

The result is 46

12. What is the output that results from the following set of PHP statements?

```
$var1 = 46.94;  
$var2 = (bool)$var1;  
print "The result is ".$var2;
```

The result is true - or - The result is 1

13. What is the output that results from the following set of PHP statements?

```
$var1 = "6 feet";  
$var2 = (int)$var1;  
print "The result is ".$var2;
```

The result is 6

14. Write the PHP function call that is equivalent to the MySQL command **use timetable;** Assume a connection to a MySQL session has been made and is named \$connection. Base your answer on a function call from the "Accessing MySQL Through PHP" list on reference sheet.

`mysql_select_db("timetable", $connection)`

15. What is the output that results from the following set of PHP statements? (4 points)

```
$stuff = array("apple" => 1, 0 => "dog", 5 => 9);  
$stuff[1] = 4.5;  
foreach($stuff as $a=>$b) echo "{$a} - {$b}\n";
```

It is important to note that for the foreach loop, the \$a represents the indexes of the array while the \$b represents the values stored in the array at that particular index. Therefore, the code is meant to output the index followed by a dash followed by the value for each element of the array. (Note: the parenthesis separate the \$a and \$b from the dash, so the output will **not** be the result of a subtraction.

Answer

```
apple - 1  
0 - dog  
5 - 9  
1 - 4.5
```

16. Describe what \$result would contain after executing the following PHP statements.

```
$myarray = array("a" => "car", "b" => "boat", "c" => "plane");  
$result = array_keys($myarray);
```

array_keys() returns an array containing the keys or indexes for the array \$myarray. Therefore, \$result is an array containing the values 'a', 'b', and 'c'.

17. What exactly does \$result contain after executing the following PHP statements.

```
$myarray = array_fill(5, 20, 0);  
$result = count($myarray);
```

count(Array) returns the number of elements in the Array. That means \$result will contain an integer representing the number of elements in \$myarray. So how big is \$myarray? The function array_fill(start, count, fill_value) creates an array where the first index equals start, the length of the array is count, and the value stored in each element of the array equals fill_value. Therefore, \$my array has 20 elements and \$result = 20.

18. What is the returned result of the PHP function **mysql_query()**? (2 points)

- a.) The next record from a table of results from a MySQL query
- b.) A reference to a table of results from a MySQL query
- c.) An array of the field names from a table in a selected MySQL database
- d.) A list of tables from a selected MySQL database

19. What is the output that results from the following set of PHP statements? (3 points)

```
function myfunc()  
{  
    static $var = 0;  
    $var += 5;  
    print "The value equals ".$var."\n";  
}  
myfunc();  
myfunc();  
myfunc();
```

The keyword *static* is used to define a single variable that is shared across all references to a function. It is only initialized once. This happens the first time the function is called. After that, it maintains its value across all function calls that any modification is carried to the next function call. Therefore, each time myfunc() is called, it is incremented by 5.

Answer

The value equals 5
The value equals 10
The value equals 15

20. In the PHP function call **\$record = mysql_fetch_array(\$result, MYSQL_NUM)**, what is the purpose of the argument **MYSQL_NUM**?

It forces the indexes of the array returned by mysql_fetch_array to be integers starting at 0.

21. After calls to `$record = mysql_fetch_array($result, MYSQL_NUM)` have exhausted all of the records from the query referenced by `$result`, what value will be returned for `$record` if one more call is made?

If there are no more records to be returned, `mysql_fetch_array` returns ***FALSE***.

The questions 22, 23, and 24 use the snippet of PHP code shown below. The numbers along the left side are line numbers and are included only as a reference. They are not part of the code.

```
1:  class myClass
2:  {
3:      var $_name;
4:      static $_num = 0;
5:      private $_id;
6:
7:      function __construct($_arg1)
8:      {
9:          $_name = $_arg1;
10:         $_num++;
11:         $_id = $_num;
12:     }
13:
14:     function printID()
15:     {
16:         print "This instance's ID is " . $_id . "\n";
17:     }
18: }
19:
20: $a = new myClass("Tommy");
21: $b = new myClass("Betsy");
22: $c = new myClass("Mary");
23: $c -> printID();
```

22. What is the purpose of the keyword "private" in line 5?

To make it so that `$_id` is only visible within the instance of the class and its functions. In other words, no code outside of the object's code can see the variable `$_id`.

23. Write the code that would retrieve the value stored in the variable `$_name` of instance `$a`.

```
$a->$_name
```

24. What is the output from line 23 going to be?

```
This instance's ID is 3
```

25. What is the difference between including a file using *include(URL_str)* and including it using *require(URL_str)*?

The function `include()` tries to continue processing the PHP script if the included file contains an error or if it doesn't exist. The function `require()` stops processing if there is an error with the included file.

SQL/MySQL Syntax:

- INSERT INTO tablename (fieldname [, fieldnames]) VALUES (value [, values])
- DELETE FROM tablename WHERE fieldname=value
- UPDATE tablename SET fieldname=value WHERE fieldname=value
- USE database
- SHOW TABLES
- DROP TABLE tablename
- SELECT [ALL | DISTINCT] *| COLUMN1[, COLUMN2] FROM TABLE1 [, TABLE2] WHERE [CONDITION1 | EXPRESSION1][AND|OR CONDITION2 | EXPRESSION2] ORDER BY fieldname [, fieldnames] [ASC|DESC]

PHP Array and String Functions:

- foreach(arrayname as [indexname =>] varname)
- array_keys(\$array_name)
- count(\$array_name)
- array_name = array_fill(integer start , integer count , mixed fill_value)
- bool sort(\$array_name)
- bool asort(\$array_name)
- string join(string delimiter, arrayname)
- array explode(string separator, string string [, int limit])
- integer strlen(string)
- string substr(source , start [, length])
- integer strpos(source , substring [, offset])

Accessing MySQL Through PHP:

- \$connection = mysql_connect ("localhost", "zabc123", "password")
- mysql_select_db("dbname", \$connection)
- \$result = mysql_query(MySQL_statement_string, \$connection)
- \$record = mysql_fetch_array(\$result [, MYSQL_NUM |MYSQL_ASSOC | MYSQL_BOTH])
- mysql_close (\$connection)
- int mysql_errno(\$connection)
- string mysql_error(\$connection)
- void exit([string or int status])