Concordia University – Centre for Continuing Education
Giuseppe Fortino (instructor)

Student full name: _____

Student ID: _____

You have **75 minutes** to complete the exam.
This exam is worth 70 points and represents 25% of the final grade for this course.

1. What is the minimal structure of a valid HTML document? (1 point)

```
<html>
        <head>
                <title>...</title>
        </head>
        <body>
        </body>
</html>
```

2. What is the purpose of the **`<title></title>`** HTML tag? (1 point)

The title is displayed in the browser tab/toolbar, in the bookmarks/favorites and is used for the link in search engine results.

3. How do you insert comments in a CSS file? (1 point)

Comments start with /* and end with */ (may be on different lines).

4. Explain the result of the following CSS directive: (2 points)

```
img { margin: 5 8px 10 0; }
```

Sets the margin around all image tags as follows: 5 pixels on top, 8 pixels on the right, 10 pixels on the bottom and no margin (0 pixels) on the left.

5. Convert the directive from 4. into four distinct directives applied to img. (2 points)

```
img {
        margin-top: 5; /* or 5px */
        margin-right: 8px; /* or 8 */
        margin-bottom: 10; /* or 10px */
        margin-left: 0; /* or 0px */
}
```

6. What does this CSS directive do? (2 points)

```
.result { padding: 2rem 4rem; }
```

Sets the padding inside elements with class *result* as follows:

padding on top and bottom = 2 times relative to base element size

padding on left and right = 4 times relative to base element size

7. What is the result of the following style definition? (2 points)

```
#mainContent p { color : red; }
```

All paragraphs inside element with id *mainContent* will have the text displayed in red.

8. Demonstrate the usage of single-line and multiple-line comments in PHP. (2 points)

```
// This is a single-line comment in PHP.

/* This is the start of a multi-line comment
...
...
… which will end here. /*
```

9. Write a PHP class which displays “MyClass class instance has been initialized!” when an object is created with this class. (3 points)

```
class MyClass
{
        public function __construct()
        {
                echo 'MyClass class instance has initialized!'."\n";
        }
}
```

10. Create a variable and assign it a new instance of the class defined in the previous question. (1 point)

```
$userclass = new MyClass;
```

11. What is the output of the following code and why? (3 points)

```
$a = '1';
$b = &$a;
$b = "2$b";
echo $a.", ".$b;
```

21, 21

Variable $b is a reference to $a, so changes made to $b are reflected in $a.
When $b is assigned “2$b”, a string, $b points to '1' (variable $a).
So “2$b” becomes “21” and gets assigned to $b.
Since $b points (refers) to $a, the content is actually stored in variable $a.
Variable $b only points to where $a is stored in memory.

Thus, $a.”, “.$b becomes “21”.”, “.”21”, hence the output 21, 21.

12. What does the `&&` operator do in PHP? (1 point)

It is the logical *and* operator.

13. Declare two string variables and display the concatenated result separated by a space. (2 points)

```
$a = "Part 1";
$b = "Part 2";
echo $a . " " . $b;
```
or
```
echo "$a $b";
```

14. What are the three (3) ways available to add styles to your web page? (3 points)

Inline:

Block inside same document:

```
<style>
    div {
        color: red;
    }
</style>
```

External file:
<link rel="stylesheet" type="text/css" href="styles.css">

15. Given the following PHP function:

```
function x($n,$a = 0,$b = 1)
{
    $result = [$a,$b];
    for($i=1;$i<$n;$i++)
    {
        $result[] = $result[$i]+$result[$i-1];
    }
    return $result;
}
```

What is returned by x(), x(11) and x(5, 55, 89) ? (5 points)

x() → returns array [0,1]
It will also trigger a warning for the missing mandatory argument $n

x(11) → returns array [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

x(5, 55, 89) → returns array [55, 89, 144. 233. 377, 610]

16. What will be the output of the code below and why? (5 points)

```
$x = 5;
echo $x;
echo $x+++$x++;
echo $x;
echo $x---$x--;
echo $x;
```

511715

Explanation:
5: $x equals 5 at first.
11: the ++ is done after evaluating the expresion, but the first $x++ must be evaluated before the +, so we have ($x++ + $x) first followed by second $x++ before returning result (5+6)
7: the previous statement incremented $x twice by 1, so from 5 to 6 to 7.
1: same as second statement but in reverse: 7-6 = 1
5: $x was 7 before decrementing by 1 twice in a row, so from 7 to 6 to 5.
Note that echo does not automatically change to new line.

17. What will be the output of each of the statements below and why? (2 points)

```
var_dump('0123' == 123);
var_dump('0123' === 123);
```

18. What will $x be equal to after the statement `$x = 3 + "15%" + "$25"` and why? (2 points)

bool (true)
Although not identical in type and value, '0123' is numerically equal to 123.

bool (false)
The identical operator === checks if the type and value is the same.
In this case, the string '0123' is not identical to the number 123.
So the condition returns false.

19. After the code below is executed, what will be the value of **`$text`** and what will **`strlen($text)`** return? Explain your answer. (3 points)

```
$text = 'John ';
$text[10] = 'Smith';
```

strlen($text) will return 11.

Because a string is an array and because we are adding 'Smith' starting at index 10, the result will be: $text = 'John S'. We will have empty array element completing up to 10.

Since we are not appending the array 'Smith', only the first character will be assigned.

The length of 'John S' is thus 11.

20. What does the follow code echo? Why? (2 points)

```
$a = "PHP";
$a = $a + 1;
echo $a;
```

The output is 1.

PHP will try to convert $a to a number in order to add 1, but because “PHP” is a string, it will not be converted (or juggleg) properly and will be ignored in the statement.
The result is $a = nothing + 1 = 1.

21. Write a function which returns 'even' for even numbers and 'odd' for odd numbers using a classic conditional statement. (3 points)

```
function EvenOrOdd($n) {
    if ($n % 2 == 0) {
        return "even";
    } else {
        return "odd";
    }
}
```

22. Write a function which returns 'even' for even numbers and 'odd' for odd numbers using a ternary conditional operator. This function should have only one line. (3 points)

```
function EvenOrOdd($n) {
    return ($n % 2 == 0) ? "even" : "odd";
}
```

23. Create an empty **`User`** class, then create an empty **`Administrator`** class which inherits **`User`**. (2 points)

```
class User {
}

class Administrator extends User {
}
```

24. Create a **`Logger`** interface which requires a log function. This function has two parameters: **`$message`** and **`$level`**. (3 points)

Create a **`MyLog`** class which implements the **`Logger`** interface. Use an **`echo`** command to mimic any actual operation. (3 points)

Write a **`Loggable`** trait with the following elements: (5 points)

- a **`$logger`** property visible to the trait and the class using the trait
- a **`setLogger`** method which accepts an input that implements **`Logger`**
- a **`log`** method which calls the logger's **`log`** method

Create a **`MyApp`** class which implements **`Logger`** interface and uses the **`Loggable`** trait. (3 points)

Finally, write a block of code that 1) creates a **`MyApp`** object, 2) sets the logger to a new **`MyLog`** instance for **`MyApp`** and 3) logs “Log is activated” with level “INFO”. (3 points)

```
interface Logger {
        public function Log($message, $level);
}


class MyLog implements Logger {
        public function Log($message, $level) {
                echo “$level: $message”; // Or something similar.
        }
```

```
trait Loggable
{
    protected $logger;

    public function setLogger(Logger $logger)
    {
        $this->logger = $logger;
    }

    public function log($message, $level)
    {
        $this->logger->log($message, $level);
    }
}


class MyApp implements Logger
{
    use Loggable;
}
```