**Bài 3: BÀI TẬP FUNCTIONS**

1. Which of the following is the correct naming convention of user defined functions?

A. &sum($var1, $var2)

B. 123\_sum($var1,$var2)

C. ^^\_sum($var1, $var2)

D. \_sum($var1, $var2)

1. Which of the following options about return statement is true?
2. You can return any type of value including arrays and objects.
3. The return statement does not return string values and Boolean values.
4. The return statement does not work with anonymous functions.
5. Q1: What is the output of the following code?

<?php

*declare*(strict\_types=1);

function multiply(float $a, float $b): int {

*return* $a \* $b;

}

$six = multiply(2, 3);

echo gettype($six);

1. Int
2. Float
3. Fatal error: Uncaught TypeError
4. Some PHP functions, like echo, do not need you to use brackets when calling them. Is this true?
5. Yes, because you can call it like this: echo "hello";
6. Yes, because echo is a special case
7. No, because echo is a language construct and not a function. All PHP functions require you to use brackets when calling them.
8. No, because all PHP functions require you to use brackets when calling them, except echo, which only requires brackets when you use more than one argument.
9. What is the output of the following code?

<?php

(function Hello() {

echo "Hello World!";

})();

1. Nothing
2. Hello World
3. An error message and "Hello World"
4. Just an error message
5. What is the output of the following code?

<?php

*declare*(strict\_types=1);

function multiply(float $a, float $b): float {

*return* (double)$a \* (double)$b;

}

$six = multiply(2, 3);

echo gettype($six);

1. Int
2. Double
3. Float
4. This generates a TypeError
5. What is the output of the following code?

<?php

function complicated($compulsory, ...$extras) {

echo "I have " . func\_get\_args() . " arguments";

}

complicated(1,2,3,4);

1. 1
2. 2
3. 4
4. This produces a notice error
5. How would you refer to the parameter with the value cat in the following function?

<?php

function complicated($compulsory, ...$extras, $animal) {

*// I want to reference the variable with the value "cat"*

}

complicated(1,2,3,"cat");

1. $animal
2. $extras[1]
3. $extras[2]
4. This produces an error
5. Consider the following PHP script:

<?php

$a = 5;

$b = 10;

function Mul()

{

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

Mul();

print($b);

?>

What can you write instead of \*\*\*\*\*\* on line 7 to get the output 50? Each correct answer represents a complete solution. Choose all that apply.

1. globals($b = $a \* $b);
2. global $a, $b;  
   $b = $a \* $b;
3. $GLOBALS['b'] = $GLOBALS['a'] \* $GLOBALS['b'];
4. $b = $a \* $b;
5. Consider the following PHP script:

<?php

$a = 5;

$b = 10;

function Mul()

{

$a =0;

$b = $a \* $b;

}

Mul();

print($b);

?>

What will be the output of the above script?

1. The script will return an error message.
2. 0
3. 10
4. 50