CS330: Programming Language Project (PLP)

Assignment 5: Functions and parameter passing

Being able to write functions that can be called more than once and sent different information each time is a huge part of most programming languages. However, there are a lot of variations in how functions are declared, where they have to be put, how they accept parameters, and how the function output is returned. In your programming language, do the following:

* Check some programming guides for your language and implement a few of the functions described there
* Write a function that takes in two numbers, multiplies them, and returns the output
* Write a recursive function (if possible)
* Write a function that takes in a string (or your language's equivalent) and splits it into two strings, then returns both strings
* Call your functions from main, and save the results of the function calls in variables.
* Write a function that tests whether your language is pass-by-reference or pass-by-value.

For your write-up, answer the following questions:

1. What is the syntax for declaring a function in your language?
   1. function functionName($parameter1, $parameter2){

do something;

return something;

}

a function can have no parameters or not return anything

1. Are there any rules about where the function has to be placed in your code file so that it can run?

Functions must be defined BEFORE calling it. For example:

#test exponential

$a = 3;

$b = 4;

$power = expo($a, $b);

echo "$a^$b= $power \n";

#EXPOPNENTIAL

#a^b

function expo($a, $b){

if($b == 1){

return $a;

}

elseif ($b == 0){

return 0;

}

else{

$c = $a \* expo($a, $b-1);

return $c;

}

}

This would not work because function was not defined before calling it.

1. Does your language support recursive functions?

Yes, PHP does. For example:

#EXPOPNENTIAL

#a^b

function expo($a, $b){

if($b == 1){

return $a;

}

elseif ($b == 0){

return 0;

}

else{

$c = $a \* expo($a, $b-1);

return $c;

}

}

If a = 2, b = 3, which is equivalent to a^b = 2^3. The recursion would do 2\*2\*2 until $b is 1.

1. Can functions in your language accept multiple parameters? Can they be of different data types?

Yes, PHP accepts multiple parameters. For example:

function multiply($a, $b) {

$c = $a \* $b;

echo "$a x $b = $c \n";

return $c;

}

PHP is not really strict in term of data types. Even if I specify the parameter as a string the beginEndFind, I still can pass an integer and it still does the job.

function beginEndFind(string $word){

$first = $word[0];

$end = substr($word, -1);

return array('first' => $first, 'end' => $end);

}

therefore, I can pass different data type as my parameters

1. Can functions in your language return multiple values at the same time? How is that implemented? If not, are there ways around that problem? What are they?

PHP does not technically return multiple values. The only way I can “cheat” is to put those values into an array. Which kinda makes sense because an array is considered as 1 value.

function beginEndFind(string $word){

$first = $word[0];

$end = substr($word, -1);

return array('first' => $first, 'end' => $end);

}

1. Is your language pass-by reference or value? Check your code against outside sources in case there is anything tricky going on (like in Perl).

Defaulty, PHP is a pass-by value. Which means the value I got inside the function and outside of the function are different, even though the parameter is the same. For example:

#a+1

function addOne ($a){

echo "a before adding 1 in the addOne function: $a\n";

$a++;

echo "a in addOne: $a\n";

}

$a = 3;

#Test a+1

echo "a in main before pass to addOne: $a\n";

addOne($a);

echo "a in main: $a\n";

#pass by value as default

As the output:

a in main before pass to addOne: 3

a before adding 1 in the addOne function: 3

a in addOne: 4

a in main: 3

The value outside of a function remains the same.

However, there is also a trick that can make PHP become pass by reference is to add & before the parameter.

#a+1

function addOne (&$a){

echo "a before adding 1 in the addOne function: $a\n";

$a++;

echo "a in addOne: $a\n";

}

As the output is going to be:

a in main before pass to addOne: 3

a before adding 1 in the addOne function: 3

a in addOne: 4

a in main: 4

1. Are there any other aspects of functions in your language that aren't specifically asked about here, but that are important to know in order to write one? What are they?

PHP does not require to state type of data of the parameter like Java does.

As always, write out the answers to these questions as though you were writing a guide for a new learner. Make your answers as clear as possible, and don't just answer the questions in one or two words. Explain the reasoning behind the answers as much as possible. If there is no clear-cut answer to a question, explain why not. And cite your sources!

Turn in both your code and your written answers, and be prepared to run your code for me in lab.