CS330: Programming Language Project (PLP) Assignment 4: Control flow

Due: October 6th

Now that you know about the data types in your programming language, it's time to start actually using them. Find out how the following control statements are written in your language, and write an example of each. If your language doesn't support these control statements specifically, try to find a way to emulate the behavior:

- **a one-condition if/else statement (i.e. "if x == true")**
- **a multi-condition if/else statement (i.e. "if x>0 && y< 10)**
- **different kinds of loops: while, do/while, for, foreach**
- **a switch-case statement**
- **break and continue statements**

For your write-up, answer the following questions:

1. What types of conditional statements are available in your language? (if/else, if/then/else, if/elseif/else). Does allow for statements other than "if" (for example, Perl has an "unless" statement, which does the opposite of "if"!)

Lua's conditional statement is if/then/else and we can also include elseif statements and nothing else.

2. Does your language use short-circuit evaluation? If so, make sure that your code includes an example.

Yes, Lua does use short-circuit evaluation, "both "and" and "or" use short-cut evaluation, that is, they evaluate their second operand only when necessary." For example, in the FizzBuzz code where "if i%3==0 and i%5==0 then", if the first condition isn't true, it won't even check the second condition.

3. How does your programming language deal with the "dangling else" problem?

The dangling else is when the lack of indentation causes confusion about whether an if or else statement is nested or not. Lua solves this problem by putting an end statement after every finished if statement or use elseif statement to continue the original if conditional statements.

4. Does your language include multiple types of loops (while, do/while, for, foreach)? If so, what are they and how do they differ from each other?

Lua includes the while, repeat until and for loops. It does not have a foreach loop. The while and repeat until loops are similar, except the condition is stated at the beginning in a while loop and at the end of a repeat until loop.

5. Can you use break or continue statements (or something similar) to exit loops?

I could use the break statements to end the loop, but there's no continue statement. However, it offers the go to statement as the same function for the continue statement.

6. If your language supports switch or case statements, do you have to use "break" to get out of them? Can you use "continue" to have all of them evaluated?

It does not support the switch or case statement, but lua can use the table function as the switch or case statement. No break or continue are required.

7. Is there anything special in terms of control flow that your language does that isn't addressed in this assignment? If so, what is it and how does it work? Make sure to include an example of it in your code as well.

If your language doesn't support anything like this, then explain why not and what it does instead (feel free to check in with me on this if you aren't sure what your language does).

This is the first time I've encountered codes that requires "then", and "end" for the for loop and while loops instead of {}

Make sure that your answers are clear, accurate, and fully-formed: remember that these tutorials are public, and GitHub users don't have the context of the assignment that you do. 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!

You can incorporate code into your tutorial to show examples, but you should also have a file in your repository that is just commented code, that demonstrates how all of these FoPC statements are written in your language. This should be a file that someone could download and run.

Works Cited:

"Lua - Loops - Tutorialspoint". ***Tutorialspoint.Com*****, 2020,** **https://www.tutorialspoint.com/lua/lua_loops.htm.**

"Programming In Lua : 3.3". ***Lua.Org*****, 2020,** **https://www.lua.org/pil/3.3.html.**