CS 330 PLP 4 Write Up:

**Question 1:** What are the Boolean values in your language?

**Answer 1:** They are true and false

**------**

**Question 2:** What types of conditional statements are available in your language? (if/else, if/then/else, if/elseif/else). Does your language allow for statements other than “if” (for example, Perl has an “unless” statement, which does the opposite of “if”!)

**Answer 2:** C++ has the classic if, if/else, else/if, and the switch statement. The switch statement allows you to pick one of multiple code blocks depending on the value of an expression. Some say that it actually works better than the if/else, that it looks cleaner.

------

**Question 3:** How does your language delimit code blocks under each condition in selection control statements?

**Answer 3**: C++ uses the curly brackets to delimit code in the control statements like if else etc.

------

**Question 4.** Does your language use short-circuit evaluation? Include an example of the short-circuit logic working or not working (or both, if your language is like Java and supports both!)

**Answer** **4:** Yes it does have short-circuit evaluation. It uses && for And and || for Or. Meaning that it will stop evaluating an expression as soon as a result can be determined. An example (in a badly written pseudocode) would look like:

If x==4 && y=4

Return “Both are true”

If x==4 || y==4

Return “atleast one is true”

-----

**Question 5.** How does your programming language deal with the “dangling else” problem?

**Answer 5:** C++ deals with the “dangling else” problem by following the rule that an else is always associated with the nearest preceding if statement that does not already have an else clause. (<https://www.geeksforgeeks.org/dangling-else-ambiguity/>)

------

**Question 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 the conditions evaluated?

**Answer 6:** C++ does support switch statements, and you do need to break to get out of them. If you don’t it will “fall through.” You can’t use continue in C++, as that is used for other things in the language.

------

**Question 7.** 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?

**Answer 7:** Yes C++ has multiple loop types. There is the while, do-while, for loop, and for each. The while loop continues executing a block of code as long as the specific condition is true. If it is false to begin with it won’t even start. The do-while loop is similar but it checks the condition after executing the loop’s body, meaning the loop will run at least once even if it is false to begin with. Also called the “post-test loop.” The for loop us used in the same way it is in python, usually used when you know how many times a loop should run, more numerical than conditional. It will run at the start and check the condition. Finally, the foreach loop is used for iterating over elements in a container like arrays or lists. It is pretty condensed and has a cleaner look to it than something more drawn out. Probably has the most simplified syntax out of all of them.

------

**Question 8.** Are loop code block variables treated differently than function code blocks?

**Answer 8:** Yes they are treated differently than function code blocks. The scope is different, as well as the lifetime. The way you declare them is different too. For code block variables they are declared/initialized only when the function is called. For the loops, they can even be declared in the loop header. They are both local to their own {} blocks.