

Livic Inoa  
Prof. Carroll  
CS-330  
October 6, 2020

## Control Flow

For C++ I have now tried out different kinds of ways to manage the control flow. C++ has if statements, else statements, and else if. When it comes to checking multi conditioned statements(&& and ||), C++ uses short-circuit evaluation. In an && statement if the first statement is false it does not check the second one, and with || statements if the first statement is true it does not check the second statement. I found that to be interesting, since it is a humanlike way to check and evaluate statements.

So far I have found that C++ uses for, while, and do/while loops. I was very interested in the do/while loop since I had not used one of those before. I think it is especially useful when it comes to avoiding errors where you are off by 1, since intuitively when I first learned while statements I always regarded the “while” as something you checked towards the end, even though I was aware it was checked at the beginning. For loops and while loops work the same way as most other languages. For loops work until a condition is met, and while loops work while a condition is met.

Break and continue statements are also able to be used in C++. Break statements completely end a loop, meanwhile continue statements simply skip over a specific segment of the loop. These were the cases at least in the code I provided. Another way to use break statements is within switch case statements. Since after each case, there is usually a break underneath. However you cannot use continue as a replacement for break within a switch case statement.