Structured Programming in Python and C++

In the previous millennium, computer code was often organized by procedural programming approach, while most modern programs are created using object-oriented programming. However, regardless of which approach is adopted, the concepts of structured programming are employed. Structured programming utilizes “control structures” such as if-statements, for-loops, and while-blocks to carry out operations in the correct order. An important distinction of structured programming is that it abolishes the “go-to” statements of earlier languages. Academic papers have been written to prove that all eventualities are covered by the constructs of structured programming, without the need for go-to statements, a source of many tricky bugs.

## Logical Operators

The control structures depend on logical expression to check whether an operation should proceed. A logical expression employs the logical operators listed below to return either true or false. For example, *a*>*b* is true if *a* is greater than *b*.

|  |  |  |
| --- | --- | --- |
|  | **C++** | **Python** |
| greater than | > | > |
| less than | < | < |
| equal to | == | ==, is |
| not equal to | != | !=, is not |
| greater than or equal to | >= | >= |
| less than or equal to | <= | <= |
| and | && | and |
| or | || | or |

## If Statements in C++

if (a<b) {  
 c=1;  
}  
else if (a>b)   
 c=2;  
}  
else {  
 c=3;  
}

## If Statements in Python

In contrast with C++, Python does NOT use brackets to encapsulate the code within an if-statements. Instead it is the indentation of code that signals whether it is part of the if-statement. Always use 4 (four) spaces per indentation. And do not forget the colons:

if a<b:  
 c=1  
elif a>b:  
 c=2  
else:  
 c=3

## For Loops in C++

for (int i=0; i<n; i++) {  
 a++;  
}

## For Loops in Python

In Python it is common to use the function “range” to run the index of a for-loop. Also observe the colon, and the use of indentation to specify which statements should be repeated:

for i in range(n):  
 a += 1  
print("a equals", a) # This statement will NOT be repeated

## While Loops in C++

while (a<b) {  
 a++;  
}

## While Loops in Python

while a<b:  
 a += 1  
print("a equals", a) # This statement will NOT be repeated