Ekaba Bisong
Programming in C++
University of Calabar

Lesson Note #29
June 03, 2015

Adapted from C++ How To Program edited for our own purposes

Logical Operators

C++ provides logical operators that are used to form more complex conditions by combining simple conditions. The logical operators are && (logical AND), || (logical OR) and ! (logical NOT, also called logical negation).

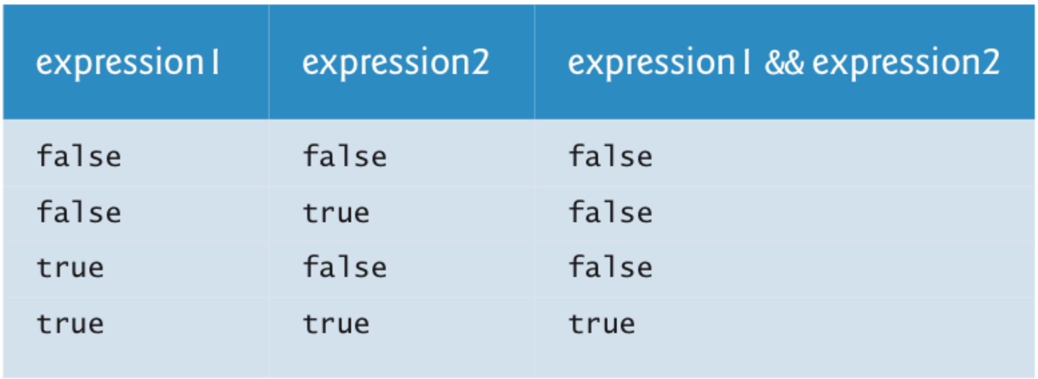
Logical AND (&&) Operator

Suppose that we wish to ensure that two conditions are both true before we choose a cer- tain path of execution. In this case, we can use the && (logical AND) operator, as follows:

```
      if ( gender == 1 && age >= 65 )
         ++seniorFemales;
```

This if statement contains two simple conditions. The condition gender == 1 is used here to determine whether a person is a female. The condition age >= 65 determines whether a person is a senior citizen. The simple condition to the left of the ***&&*** operator evaluates first. If necessary, the simple condition to the right of the ***&&*** operator evaluates next.

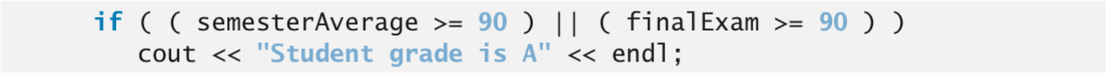
The table summarizes the && operator and shows all four possible combinations of false and true values for expression1 and expression2. Such tables are often called truth tables. C++ evaluates to false or true all expressions that include relational opera- tors, equality operators and/or logical operators.



expression1	expression2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

Logical OR (||) Operator

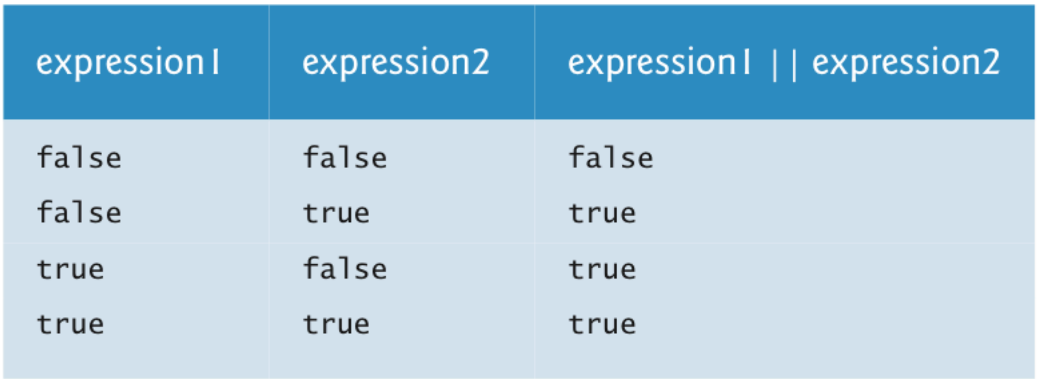
Now let's consider the || (logical OR) operator. Suppose we wish to ensure that either or both of two conditions are true before we choose a certain path of execution. In this case, we use the || operator, as in the following program segment:



```
if ( ( semesterAverage >= 90 ) || ( finalExam >= 90 ) )
   cout << "Student grade is A" << endl;
```

The if statement considers the combined condition and awards the student an "A" if either or both of the simple conditions are true. The message "Student grade is A" prints unless both of the simple conditions are false.

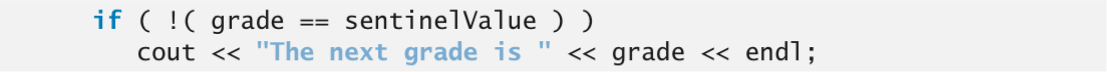
The table summarizes the || operator



expression1	expression2	expression1 || expression2
false	false	false
false	true	true
true	false	true
true	true	true

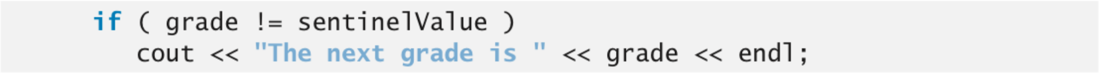
Logical Negation (!) Operator

C++ provides the ! (logical NOT, also called logical negation) operator to "reverse" a condition's meaning. The unary logical negation operator has only a single condition as an operand. The unary logical negation operator is placed before a condition when we are interested in choosing a path of execution if the original condition (without the logical negation operator) is false, such as in the following program segment:



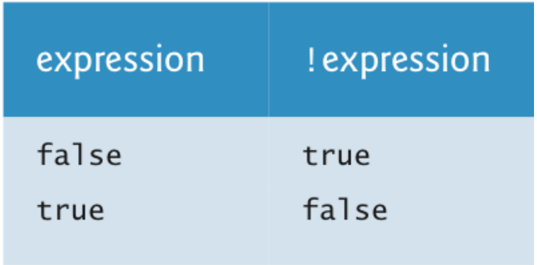
```
if ( !( grade == sentinelValue ) )
   cout << "The next grade is " << grade << endl;
```

or



```
if ( grade != sentinelValue )
   cout << "The next grade is " << grade << endl;
```

The table summarizes the Negation (!) operator



expression	!expression
false	true
true	false