# CIS7 Lab 2: Using Logical Operators and Boolean Variables

In this lab, we will use bitwise, logical operators and Boolean variables in C++ programs.

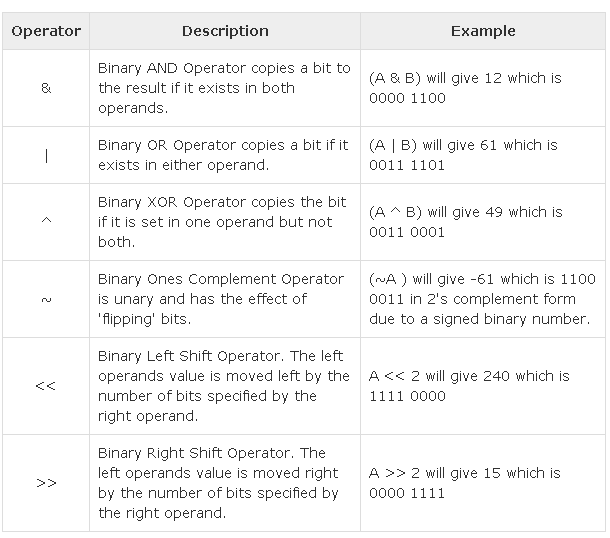
## Bitwise Operators in C++:

Assume if A = 60; and B = 13; now in binary format they will be as follows −

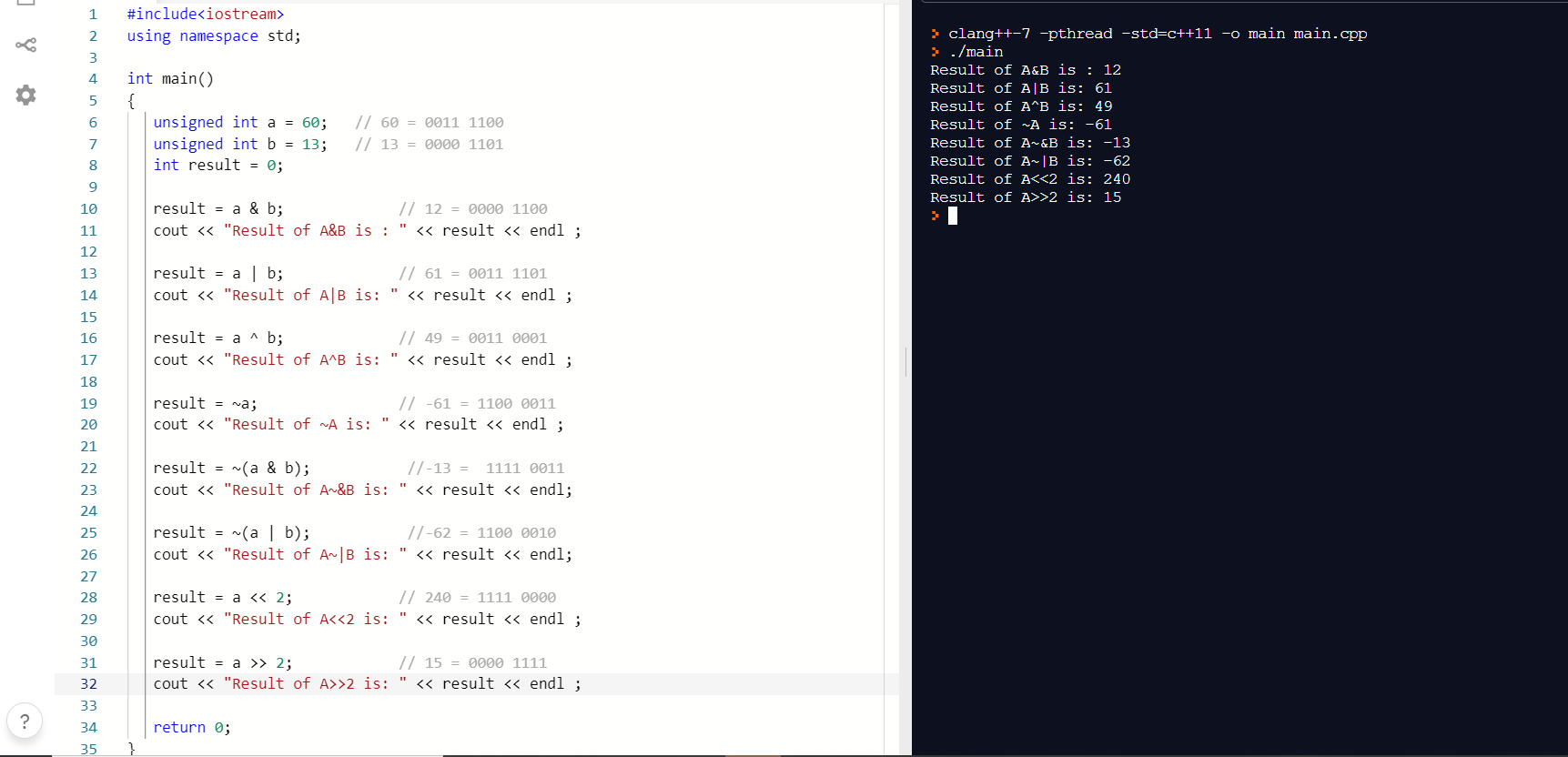
A = 0011 1100 B = 0000 1101

A&B = 0000 1100 A|B = 0011 1101

A^B = 0011 0001 ~A = 1100 0011



**Example 1: Bitwise operator program**



1. Refer to the above example 1. Given **P = 40** and **Q = 17**, write a C++ program using bitwise operators to determine the result, R, for the following:
2. **P & Q**
3. **P | Q**
4. **P ^ Q**
5. **~P**
6. **~(P &Q)**
7. **~(P | Q)**
8. **P <<3**
9. **P>>3**

Provide screen capture of the output and code.

A screenshot of a computer

Description automatically generated

\*Note: Instruction to take screen capture on macOS see <https://support.apple.com/en-us/HT201361>. For Windows PC – press Ctrl+PrtSc button on keyboard or use Snipping Tool, then paste it to MS Word.

## Boolean Variables

**Boolean variables** are variables that can have only two possible values: **true (1), and false (0).**

To declare a Boolean variable, we use the keyword **bool**.

To initialize or assign a true or false value to a Boolean variable, we use the keywords **true** and **false**.

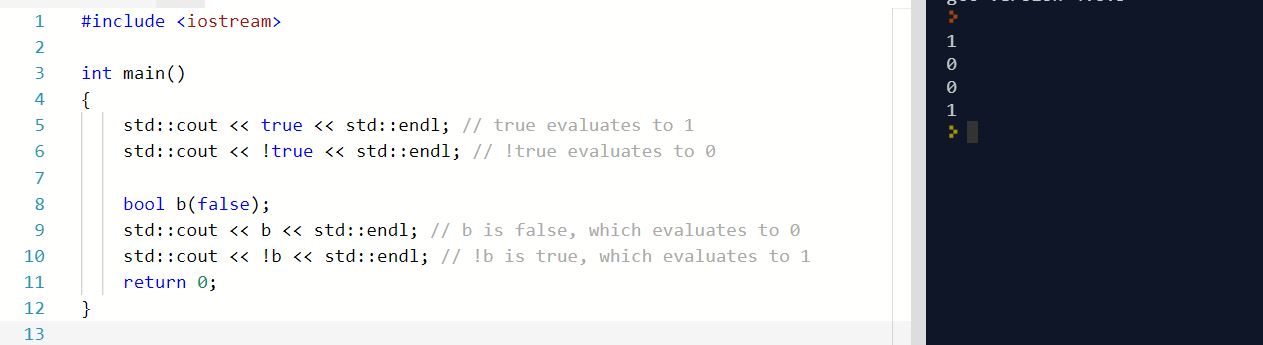
|  |  |
| --- | --- |
|  | bool b1 = true; // copy initialization  bool b2(false); // direct initialization  bool b3 { true }; // uniform initialization (C++11)  b3 = false; // assignment |

Just as the unary minus operator (-) can be used to make an integer negative, the logical NOT operator (!) can be used to flip a Boolean value from **true** to **false**, or **false** to **true**:

|  |  |
| --- | --- |
|  | bool b1 = !true; // b1 will have the value false  bool b2(!false); // b2 will have the value true |

Boolean values are not actually stored in Boolean variables as the words “true” or “false”. Instead, they are stored as integers: true becomes the integer 1, and false becomes the integer 0. Similarly, when Boolean values are evaluated, they don’t evaluate to “true” or “false”. They evaluate to the integers 0 (false) or 1 (true).

**Example 2: program using Boolean Variables to evaluate TRUE and !TRUE**



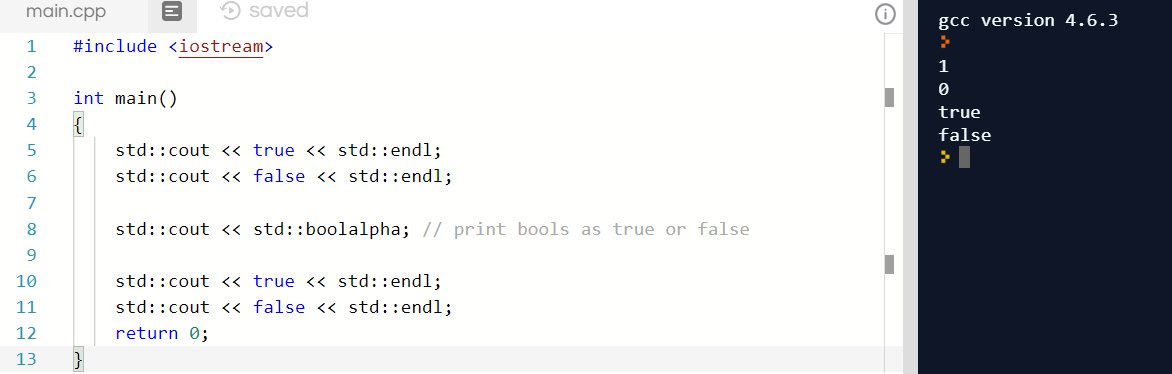
1. Using the below example, create a program in C++ that evaluates the **FALSE** and **!FALSE** values. Then, include the **bool** variable to evaluate the **TRUE** and **!TRUE** value of the variable. Provide screen capture of the output and code.

A screenshot of a computer

Description automatically generated

## boolalpha

**Example 3: std::boolalpha** is used to print “true” or “false” instead of 0 or 1 as shown in below example:

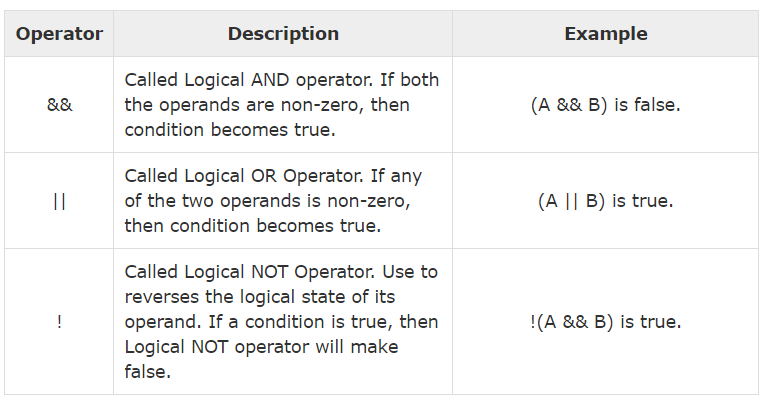


1. Create a C++ program that Illustrates the use of **std::boolalpha** to print **TRUE** bools value only. Provide screen capture of the result and code.

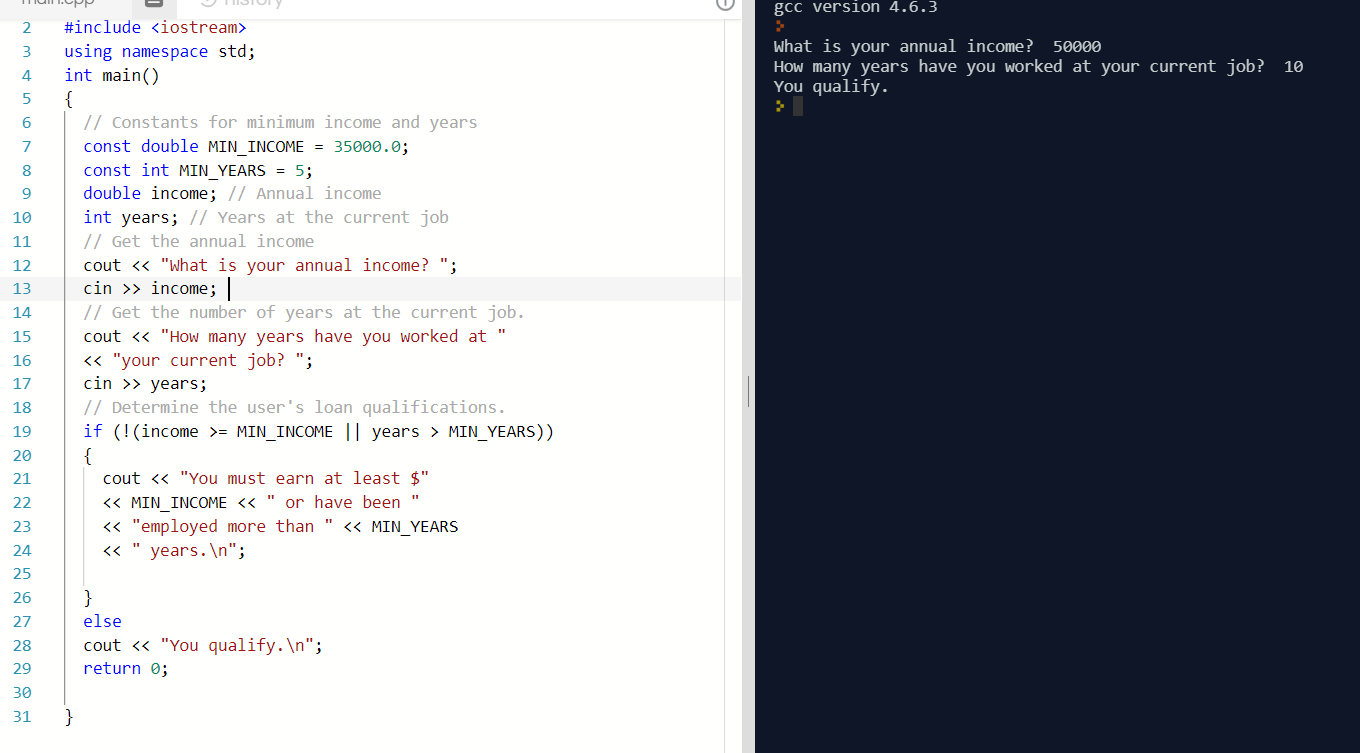
A screenshot of a computer

Description automatically generated

## Logical Operators in C++



**Example 4: C++ program that uses logical NOR operator.**

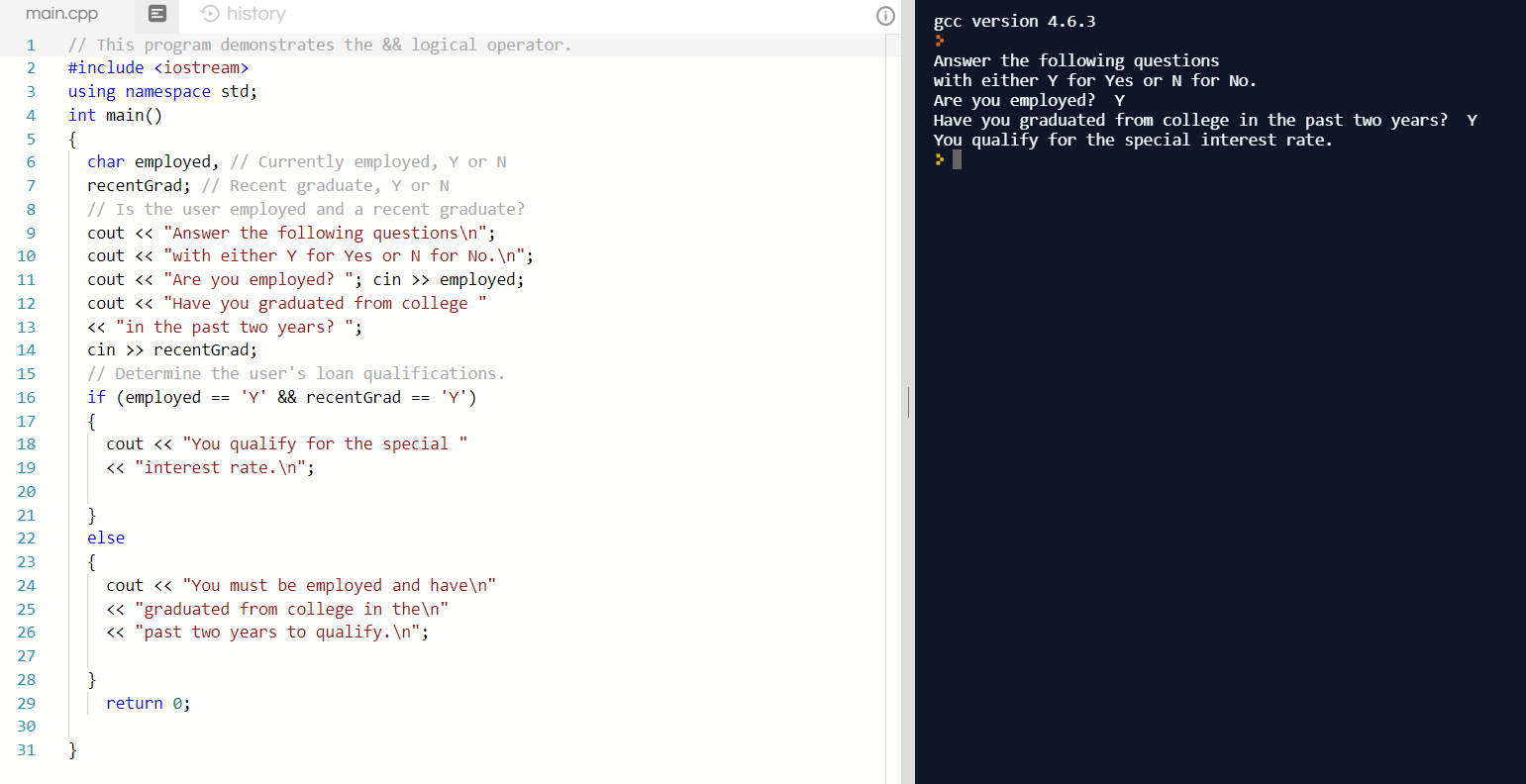


1. In the above example Logical NOR Operator is used in the condition to disqualify loan applicants based on minimum annual income, years of current employment. Refer the example 4 and create a program use the **NOR operator !(||)** to determine if a **student has earned an associate degree by completing 60+ credits or 5+ semesters as a full-time student**. Provide screen capture and code.

A screenshot of a computer screen

Description automatically generated

**Example 5: C++ program using logical && operator to qualify for loan special interest rate.**



1. In the above example 5, Logical AND operator (&&) is used to check criteria for special loan interest rate. **Refer to the example 5 and create a program that uses Logical AND operator to check if the user meets the following criteria for car insurance discount:**
2. **older than 21 and**
3. **has been driving for 5+ years**

Provide screen capture and code.

A screenshot of a computer screen

Description automatically generated

Submit a document (.docx or .pdf) that contains Lab 2 screen captures of code and output in Canvas.