
<Company Name>

Boolean Logic Simulator

User's Manual

Version <1.1>

Boolean Logic Simulator in C++	Version: <1.0>
User's Manual	Date: 04/29/2024
UM	

Revision History

Date	Version	Description	Author
04/29/2024	1.0	Initial Template Filled Out	Kevinh Nguyen
04/30/2024	1.1	Finished troubleshooting, examples, and FAQ section; minor template changes	Raj Kaura

Boolean Logic Simulator in C++	Version: <1.0>
User's Manual	Date: 04/29/2024
UM	

Table of Contents

1. Purpose	4
2. Introduction	Error! Bookmark not defined.
3. Getting started	Error! Bookmark not defined.
4. Troubleshooting	4
5. Example of uses	4
6. Glossary	Error! Bookmark not defined.
7. FAQ	5

Boolean Logic Simulator in C++	Version: <1.0>
User's Manual	Date: 04/29/2024
UM	

Test Case

1. Purpose

The purpose of this manual is to provide a solid foundation for users to understand and utilize this program properly. It aims to provide clear instructions, troubleshoot common problems, and offer examples of practical uses.

2. Introduction

Welcome to HL Group's Boolean Logic Simulator in C++! We are so happy you could be here with us. This software is designed to parse and evaluate Boolean expressions like AND, OR, XOR, and much more! Thank you to Kevinh Nguyen, Changwen Gong, Raj Kaura, Riley England, Abdulahi Mohamed, and Kemar Wilson for developing this project!

3. Getting started

To use the Boolean Logic Simulator, follow these steps:

- **Installation:**
 - Download the source code from the [Github Repository](#)
 - github.com/KNETernity/348HL
 - Compile the program using a C++ compiler (e.g., g++)
 - Execute the compiled file
- **Entering Expressions:**
 - Input your desired Boolean expression when prompted
 - Use valid Boolean operators

4. Troubleshooting

If you're encountering an unexpected error, check below for possible solutions to common errors:

1. **Missing operand:**
If you miss an operand the program will not evaluate the expression correctly. To ensure this doesn't happen, double check your input and whether it makes sense.
2. **Unknown operator:**
Make sure to use the correct symbols for each Boolean operator. A list of the operators is shown in section 5 below.
3. **Parentheses mismatched:**
Ensure that every parenthesis is closed and that there aren't any additional ones as this will result in your expression being evaluated incorrectly or may even result in an error.
4. **Invalid characters:**
In this program, only T and F are valid truth values. Do not use variable characters like A, B, etc as the program is only initialized for T and F truth values.
5. **Operator after operand:**
Ensure that your expression has operators like NOT before any truth values. If they're after, the program will not be able to parse them.

5. Examples

Examples of fundamental logic below.

1. **AND (&)**
Expression: T & T
Evaluation: True

Boolean Logic Simulator in C++	Version: <1.0>
User's Manual	Date: 04/29/2024
UM	

2. OR (|)
Expression: T | F
Evaluation: True
3. NOT (!)
Expression: !(T | F)
Evaluation: False
4. NAND (@)
Expression: T @ T
Evaluation: False
5. XOR (\$)
Expression: T & F
Evaluation: True

6. Glossary of terms

1. Boolean Logic Simulator

Program that parses through and evaluates Boolean expressions, handling operators like AND, OR, NOT, NAND, and XOR and evaluating them in accordance with Boolean algebra rules.

2. GitHub Repository

Online platform where code of the Boolean Logical Simulator is located. Link is located in section 3.

3. C++ Compiler

Programming tool that converts the source code into an executable that operates the Boolean Logic Simulator

4. Order of Evaluation

This is the order in which the operators are evaluated. For the operators we are using, the order is parentheses, NOT, AND, NAND, XOR, and OR from left to right.

5. Invalid Expression

An invalid expression is one that is not able to be processed. Refer to troubleshooting (section 4) above to see examples of this.

6. Terminal

Program that allows user to interact with a computer system, typically via a command-line interface (CLI). In this Boolean Logic Simulator, we input expressions via the CLI and receive the evaluation via the CLI.

7. FAQ

If you have a question, refer below.

1. What will happen if my expression contains invalid character(s)?
This will result in an error message being displayed and the program ending. Make sure to include only valid operators and truth values in your expression.
2. Can I define custom variables and run the program that way?
Currently, you cannot do this as the program only supports T and F. We hope to add this feature in the future.
3. Can the Boolean Logic Simulator handle a very large input?
Yes, the Boolean Logic Simulator can handle inputs of any size. However, there may be an increase in compilation time due to this.