Kung-Fu Programmers

Boolean Expression Evaluator

User's Manual

Version 1.0

Revision History

Date	Version	Description	Author
01/May/24	1.0	Initial publication.	Schmidt, S., Stonestreet, B., Rodenberg, S., Medallada, S., Whitmer, K.

Table of Contents

User Manual

1. Purpose

This User Manual for the Boolean Expression Evaluator will explain how to use the product to its fullest potential. Thank you for choosing Kung-Fu Programmers for your Boolean Expression evaluation needs.

2. Introduction

The Boolean Expression Evaluator is a sophisticated software tool designed to process and evaluate expressions utilizing Boolean logic. It supports basic Boolean operations such as AND, OR, NOT, XOR, and NAND. Additionally, the evaluator facilitates the use of user-defined variables and parenthesis, ensuring robust handling of complex expressions.

Throughout the development of the evaluator, the Kung-Fu Programmers have prioritized delivering an intuitive interface that simplifies navigation and application, making the software accessible to users of varying technical backgrounds. The Boolean Expression Evaluator's straightforward user interface, combined with its comprehensive operational capabilities, makes it an invaluable resource for students, developers, and anyone needing efficient and accurate evaluations of logical expressions.

Supported Features:

- Basic Boolean Logic Operations
- Complex Boolean Logic Expressions
- Defining Variables for Boolean Values
- User-Friendly Interface / Intuitive Design

Install and Run:

To Install the Boolean Expression Evaluator, you will need to download the source code and compile it on your machine. Follow the steps below to complete this process.

- Visit the GitHub site here: https://github.com/Bstoney15/TermProject.git
- Click on "Code", then click on "Download Zip".
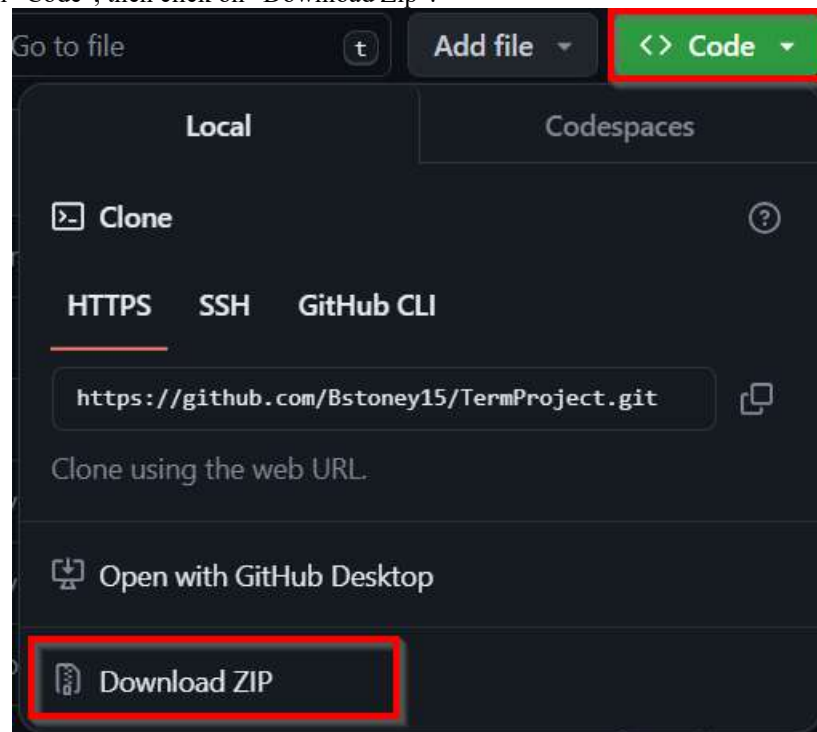
- Go to your Downloads folder, this location differs depending on your operating system. For Windows, this is in your File Explorer.
- Extract the contents of the TermProject-main.zip file.

- Once you have extracted the folder, find the extracted folder and right click it.
- From there you should see "Open in Terminal". Click it.
- Once you have opened the folder in the terminal, run "CD src". This will then open the source file in the terminal.
- The last step before you can run the exe is to type "Make" into the terminal. (note you must have g++ installed on your computer to run this command)
- Now just type "main" in the terminal and the program will run inside of the terminal.

(There is also a precompiled version in the src folder, but it is compiled for windows.)

3. Getting started

Basic Usage:

- ▪ **Launch Software:**

  - ▪ Run the Boolean Expression Evaluator program.
  - ▪ System's operation selection menu will display.

```
  - Boolean Expression Evaluator Menu -
enter a letter to select an option

N. New Expression
E. Evaluate Expression
D. Define a variable
S. Show current variables
R. Remove an existing variable

○ Enter anything else to exit
  - Enter selection letter:
  ▯
```

- ▪ **Menu Navigation and Operations:**

  User input for operation selections is not case sensitive

- ▪ Operators Key:

  Boolean Expression Evaluator's Operators to Boolean Logic Operations Conversion Table:

Boolean:	**AND**	**OR**	**NOT**	**NAND**	**XOR**
Evaluator:	**'&'**	**'|'**	**'!'**	**'@'**	**'$'**

Boolean Values:

  - ▪ 'T' = True
  - ▪ 'F' = False

- ▪ Enter a new expression:

  - ▪ Type "N" in the console and press enter to be prompted for a new expression.
  - ▪ Display will change to expression prompt, "Enter a new expression: "
  - ▪ After entering a new expression, the user is asked if they would like to have the expression evaluated. The user can type 'Y' for yes or 'N' for no and enter.

```
Enter a new expression: T
Evaluate the expression? (Y/N): ▯
```

  - • After entering 'Y' following the evaluation prompt, the result of the expression's evaluation will be displayed, or an error notification will appear indicating it could not be evaluated. Press and enter any key to return to the operations menu.
  - • After entering 'N' (selecting not to evaluate input), the menu will return to the display.
  - • Any new expression entered in the system will become the "current expression."

```
Current Expression: T&F

N. New Expression
E. Evaluate Expression
D. Define a variable
S. Show current variables
R. Remove an existing variable

Enter anything else to exit
 - Enter selection letter:
▯
```

- Evaluate current expression:

  - From the main menu, enter 'E' to see the results of the system's evaluation of the current expression.

```
Expression = T&F
Result = F

Press any key to continue...▯
```

  - From the evaluation results display console, press and enter any key to return to the menu.

- Define a new variable:

  - From the system's main display menu, enter 'D' into the console to select to define a new variable and assign it a value.
  - After entering 'D,' the user will see the display prompt: "Enter the variable character: ," all user-defined variables must be named as a single letter.

```
Enter the variable character:
x
Enter the variable value:
T▯
```

  - After entering a variable, the user will be prompted to enter a corresponding value. For the evaluation of any successive expressions. All variable values must be a Boolean value of 'T' or 'F,' otherwise an error will be raised because the evaluator will not be able to determine a Boolean value as a result.

- Show currently defined variables:

  - From the operations menu users can enter 'S' into the console to opt for requesting from the system a display of the currently defined variables.

```
Current Variables:

x = T
y = F
z = F

press any key to continue...
▯
```

  - While viewing the current variables display, press, and enter any key to return the menu.

- ▪ Remove a current variable:

    - ▪ From the menu, enter 'R' to select the variable removing operation.
    - ▪ After selecting to remove a variable, users will be presented with the display of the currently defined variables and prompted for which one they would like to remove.


```
current Variables:

y = F
z = F

Enter the variable character:
```

    - ▪ The user must enter a current variable, otherwise they will be notified that the variable they have entered is not defined.

4. Advanced features

- **Variables Handling:**

  - Users can define new variables and assign corresponding Boolean value ('T' or 'F'), variables must be named single letters. There is no definite limit to the number of variables that can be added in each session.
    - Enter 'D' from the main menu to be prompted for a new variable and value.

  - Defined variables are saved in a list structure for the duration of the Boolean Expression Evaluator's current runtime. Current variables have been defined and will be used as needed in expression evaluation processes.
    - Enter 'S' from the main menu to see a display of the currently defined variables.

  - Variables can be removed from the current variables list.
    - Enter 'R' from the main menu to show the current variables and be prompted for which one to remove.

- **Expression Handling:**

  - Newly entered Boolean expressions will be temporarily saved until the evaluator session is stopped, or a new expression is created.
  - Each expression entered is set as the "current expression," which will be displayed above the main menu.
    - Enter 'N' from the main menu to be prompted to input a new expression.

  - After Entering a new expression, user will have the option of immediately requesting its evaluation or return to the menu instead.
    - For instance: in the case of current variable modification may be required for the expression's evaluation.

  - The current expression can be evaluated at any time from the main menu of the evaluator. This makes it convenient for users to apply alterations to the current variables and values to analyze logic expression behavior under varying circumstances.
    - Enter 'E' from the main menu to request the current expression to be evaluated.

- **Error Handling:**

  - When a processing error occurs during the system-user interaction, an error flag will be created to indicate there is a problem with inputs or requests.
  - Error flags are given an integer value that corresponds to various kinds of errors, this allows for specialized handling of errors depending on their cause. If an error occurs a description of the error type will be displayed to notify the current software user.

- **Error Flags Key:**

Error Type:	Flag Value:
Operator error	1
Variable error	2
Value error	3
Parentheses error	4
NOT operation misuse	11
AND operation misuse	12
NAND operation misuse	13
OR operation misuse	14
XOR operation misuse	15
First or last character is an operand	16
Undefined variable	21
True and false back-to-back	31
Parentheses closed before it was opened	41
Parenthesis was never closed	42

5. Troubleshooting

Error Messages:

The Boolean Expression Evaluator's output package handles the interpretation of error flags and constructs a corresponding error notification. The error notification will be displayed to the user with a brief description of the error (eq. parentheses syntax error, etc.). The software's error notification system allows users to quickly identify what is causing the error and efficiently fix the problem.

Operation Misuses:

Avoid syntax errors by ensuring expressions are comprised of logic operators placed correctly between Boolean values, variables, or composite expressions.

6. Examples

- Select "N" to enter a new expression.

```
N. New Expression
E. Evaluate Expression
D. Define a variable
S. Show current variables
R. Remove an existing variable

Enter anything else to exit
  - Enter selection letter:
N
```

- Select "Y" to evaluate the expression when you enter it.

```
Enter a new expression: T&T
Evaluate the expression? (Y/N): Y
```

6.1   Example 1:

```
Enter a new expression: (T|F)$F
Evaluate the expression? (Y/N): Y

Expression = (T|F)$F
Result = T
```

6.2   Example 2:

```
Enter a new expression: (F@T)|(T@F)
Evaluate the expression? (Y/N): Y

Expression = (F@T)|(T@F)
Result = T
```

6.3   Example 3:

```
Enter a new expression: (T$T)&F
Evaluate the expression? (Y/N): Y

Expression = (T$T)&F
Result = F
```

6.4   Example 4:

```
Enter a new expression: ((F@T)$(T|(F&F)))&(T&(T@(!T)))
Evaluate the expression? (Y/N): Y

Expression = ((F@T)$(T|(F&F)))&(T&(T@(!T)))
Result = F
```

7. Glossary of terms

- **Boolean:**
  - A binary variable having one of two values, eq. "True" or "False."

- **Expression (Boolean Expression):**
  - A combination of Boolean values (true or false), Boolean variables, and operators can be evaluated to result in a single Boolean value.

- **Operator:**
  - A symbol indicating a corresponding logical operation on one or more operands.

8. FAQ

Frequent questions and answers:

- Q: Can I use lower case for Boolean values?

  A: Yes, the Boolean Expression Evaluator's input processes are not case sensitive.

- Q: What happens if I enter an incorrect expression?

  A: An error notification will be displayed, from which you will be able to return to the evaluator's menu.

- Q: Can I save expressions to evaluate later?

  A: User defined expressions and variables are only stored for the current session.