

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

ECE 150 *Fundamentals of Programming*

The ternary or conditional operator

ECE150

Douglas Wilhelm Harder, M.Math.
Prof. Hiren Patel, Ph.D.
Prof. Werner Dietl, Ph.D.

© 2020 by the above. Some rights reserved.

CC BY NC SA

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 2

Outline

- In this topic, we will
 - Review conditional statements
 - Describe the conditional operator
 - Usually just called “the” ternary operator
 - Look at examples
 - Recommend the use of comments and parentheses

CC BY NC SA

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 3

Conditional statements

- To date, we have seen unary and binary operators
 - The conditional operator takes three operands
 - Consequently, it is often just referred to as “the” ternary operator
- We have seen conditional statements, but both the consequent and alternative bodies must be separate statements


```
if ( condition ) {
    // consequent body
} else {
    // alternative body
}
```

CC BY NC SA

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 4

The conditional operator

- The ternary operator works as follows:
condition ? consequent-expression : alternative-expression
- If the condition is true, the operator evaluates to the consequent expression otherwise, the operator evaluates to the alternative expression
- The ternary operator can be used wherever the expressions would be appropriate

CC BY NC SA

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 5

The conditional operator

- For example:

```
double abs( double x ) {
    return ( x >= 0.0 ) ? x : -x;
}

double sinc( double x ) {
    return ( x != 0.0 ) ? (std::sin(x)/x) : 1.0;
}
```



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 6

The conditional operator

- To see how it can be used in an arithmetic expression:

```
int main() {
    double x{};
    double y{};
    double diff{};
    std::cout << "Enter a value of x: ";
    std::cin >> x;
    std::cout << "Enter a value of y: ";
    std::cin >> y;

    diff = ((x >= y) ? x : y) - ((x <= y) ? x : y);

    std::cout << "|x - y| = " << diff << std::endl;

    return 0;
}
```



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 7

The conditional operator

- Suppose you are converting a double to an int:

```
int main() {
    double x{};
    unsigned int n{};

    while ( true ) {
        std::cout << "Enter a positive double: ";
        std::cin >> x;

        Enter a positive real: 3.14
        if ( x >= 0.0 ) {
            break;
            Enter a positive real: 9876543210
        }
        4294967295

        n = (x >= 4294967295.0) ? 4294967295 : x;
        std::cout << n << std::endl;

        return 0;
}
```



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 8

The conditional operator

- Note that you don't have to remember or recalculate $2^{32} - 1$:

```
#include <iostream>
#include <limits>

int main() {
    int main() {
        std::cout << "Min int: "
        << std::numeric_limits<int>::min() << std::endl;
        std::cout << "Max int: "
        << std::numeric_limits<int>::max() << std::endl;
        std::cout << "Min unsigned int: "
        << std::numeric_limits<unsigned int>::min() << std::endl;
        std::cout << "Max unsigned int: "
        << std::numeric_limits<unsigned int>::max() << std::endl;

        return 0;
}
```

Output:
Min int: -2147483648
Max int: 2147483647
Min unsigned int: 0
Max unsigned int: 4294967295



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 9

The conditional operator

- Suppose you are trying to avoid a division by zero:

```
int main() {
    unsigned int m{};
    unsigned int n{};

    std::cout << "Enter a non-negative integer: ";
    std::cin >> m;

    std::cout << "Enter another non-negative integer: ";
    std::cin >> n;

    int result{ (n == 0) ? std::numeric_limits<unsigned int>::max()
                : (m/n) };

    std::cout << "m/n = " << result << std::endl;

    return 0;
}
```



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 10

The conditional operator

- Suppose you are trying to avoid a division by zero:

```
int main() {
    int m{};
    int n{};

    std::cout << "Enter an integer: ";
    std::cin >> m;

    std::cout << "Enter another integer: ";
    std::cin >> n;

    int result{ (n == 0) ? ((m < 0) ? -4294967295 : 4294967295)
                : (m/n) };

    std::cout << "m/n = " << result << std::endl;

    return 0;
}
```



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 11

Comments and parentheses

- The conditional operator is not *natural* for most programmers
 - It is beneficial to comment any ternary operator that is more complex than just evaluating to one statement
- If any of the operands are any more complex than
 - A local variable, parameter, function call or literal
 - One of these with a unary operator
 put parentheses around them
- If the conditional operator is being used in an algebraic or logical expression,
 - put parentheses around the entire operator and its operands
 - If it is the right-hand side of an assignment, parentheses are not needed



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering

Writing comments 12

Summary

- Following this lesson, you now:
 - Understand the C++ conditional or “ternary” operator
 - Know how to use it
 - Understand you should be careful with it:
 - Use comments and parentheses to make your intentions clear



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 13

References

- [1] Wikipedia: <https://en.wikipedia.org/wiki/%3F>:



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 14

Acknowledgments

None so far.



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 15

Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using Consolas.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

<https://www.rbg.ca/>

for more information.



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING
Department of Electrical and Computer Engineering

Writing comments 16

Disclaimer

These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.

