



# Logical (Boolean) Operators

Instructor: Chi Tse (Ricky)

# Logical Operators



<b>SYMBOL</b>	<b>OPERATION</b>	<b>C++ SYNTAX</b>	<b>ACTION</b>
&&	AND	<i>expr1 &amp;&amp; expr2</i>	Evaluate <i>expr1</i> . If it's true, evaluate <i>expr2</i> . Then, if both are true, return true; otherwise, return false.
	OR	<i>expr1    expr2</i>	Evaluate <i>expr1</i> . If false, evaluate <i>expr2</i> . Then return true unless both are false.
!	NOT	<i>! expr1</i>	Evaluate <i>expr1</i> . Reverse the true/false value.

(from chapter 2 of textbook)

# The “and” Operator &&

- We can combine two comparisons using the “and” operator.
- The symbol for “and” operator is &&.
- Syntax for a Boolean Expression using &&  
*(comparison\_1) && (comparison\_2)*
- The Boolean expression is true only if both comparisons are true.

# The “and” Operator &&

Example: With an *if – else* Statement

```
If ( (score > 0) && (score < 10) )  
    cout << “The score is between 0 and 10 \n”;  
else  
    cout << “The score is not between 0 and 10 \n”;
```

# The “or” Operator ||



- We can also combine two comparisons using the “or” operator.
- The symbol for “or” operator is ||
- Syntax for a Boolean Expression using ||

*(comparison\_1) || (comparison\_2)*

- The Boolean expression is true if one or both of the comparisons are true.

# The “or” Operator ||



Example: With an *if – else* Statement

```
if ( (x == 1) || (x == y) )  
    cout << “x is 1 or x equals to y.\n”;  
else  
    cout << “x is neither 1 nor equal to y. \n”;
```