

- Used with Boolean expressions
- Not makes a False expression True and vice versa
- And will yield a True if and only if both expressions are True
- Or will yield a True if at least one of both expressions are True



To test if *n* falls between 2 and 5:

(2 < n) And (n < 5)



The following is NOT a correct way to test if *n* falls between 2 and 5:

(2 < n < 5)



n = 4, answ = "Y" Are the following
expressions true or false?
Not (n < 6)
(answ = "Y") Or (answ = "y")
(answ = "Y") And (answ = "y")
Not(answ = "y")</pre>



- The order of operations for evaluating Boolean expressions is:
- 1. Arithmetic operators
- 2. Relational operators
- 3. Logical operators



Arithmetic Order of Operations

Parenthesis
 Exponentiation
 Division and multiplication
 Addition and subtraction



They all have the same precedence



Not
 And
 Or



- A condition is an expression involving relational and/or logical operators
- Result of the condition is Boolean that is, True or False



- A common error is to replace the condition *Not (2 < 3)* by the condition (2 > 3)
- The correct replacement is (2>=3)
 because >= is the opposite of <, just as
 <= is the opposite of >



- A variable declared with a statement of the form Dim var As Boolean
- is said to have Boolean data type. It can assume just the two values True and False.
- Example:
 - Dim boolVar As Boolean

boolVar = 2 < 6

txtBox.Text = boolVar