# Relational Operators and the If Statement

## Conditionals

So far, we can Input, Output and Calculate

 How can we explore relationships between data?

 How can our program only do things sometimes?

## Decisions!

- Relational Expressions allow our program to make a decision
  - based on the data in the program

 What are some decisions we might want out program to make?

## Relational Expression

- Relational expression: an expression that uses a Relational Operator
  - its value is a Boolean value (True or False)

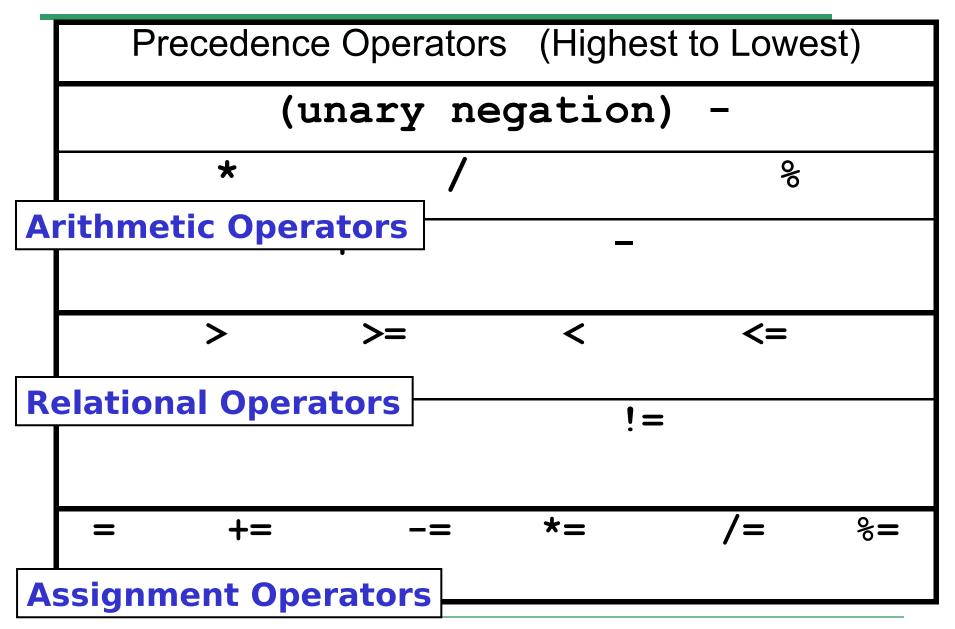
```
int height=32;
const int MIN_HEIGHT =34;
height > MIN_HEIGHT
height == MIN_HEIGHT // ==
height >= (MIN_HEIGHT - 2)
```

# Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

# Precedence (page 1101)



 What is the value of the following Relational Expressions?

```
int width = 99, height = 42;
width > height
-width <= height</pre>
width != height
width == (width + 1)
width == width + 1
```

Relational Operators work on Integers, Floating point numbers, and Characters.

## The if Statement

- We execute each statement in our program in order.
- What if we only want to execute a statement sometimes?
- The if Statement!

```
if ( condition )
{
   //statements
}
```

## Practice: What is the output?

```
int age;
const int VOTING AGE = 18;
cin >> age;
if ( age >= VOTING AGE )
  cout << age << " > "
   << VOTING AGE;
  Cout << " You can vote!"
    << endl;
```

- For the problem below:
  - what data will you need?
  - what will you need to do conditionally?
    - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

"You passed!"

"You failed!"

# Boolean value (True or False)

 How does the computer represent True and False?

```
bool value;
int x = 5, y = 10;
value = x > y; // value = ??
value = x == y; // value = ??
value = x == y - 5; // value = ??
// how does this output look?
cout << "Value is: " << value;</pre>
```

 What C++ statement would we write make the following determinations?

```
bool value;
int yourAge = 22,
  currentYear = 2009;
```

- Where you born before 1990?
- Is you age evenly divisible by 7?

# Coding Standards

```
If you only have ONE
if( expression
                          statement in the body
                           of the if, the { } are
                           optional in C++.
   statement 1;
                       \longrightarrow For this class, the \{ \}
                           must ALWAYS be used.
                           Not using { } will result
      expression
                          In a loss of style points.
   statement 1;
                          The { } must also be on
                           their own line.
                           Why?
```

### More on Truth

```
    Non-zero is

int x = 5, y = 0;
                        considered true
if (y - x)
 // This will be executed
  cout << "y - x is True" << endl;</pre>
if (y)
 // This will NOT be executed
  cout << "y is True" << endl;</pre>
```