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 our program to make?
Relational Expression

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

```cpp
int height = 32;
const int MIN_HEIGHT = 34;

height > MIN_HEIGHT
height == MIN_HEIGHT  // ==
height >= (MIN_HEIGHT - 2)
```
# Relational Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>==</td>
<td>Equal to</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>

- All are binary operators
- Left to right associativity
### Precedence Operators (Highest to Lowest)

<table>
<thead>
<tr>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>(unary negation)</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>/</td>
</tr>
<tr>
<td>%</td>
</tr>
</tbody>
</table>

#### Arithmetic Operators

<table>
<thead>
<tr>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
</tr>
<tr>
<td>&gt;=</td>
</tr>
<tr>
<td>&lt;</td>
</tr>
<tr>
<td>&lt;=</td>
</tr>
</tbody>
</table>

#### Relational Operators

<table>
<thead>
<tr>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
</tr>
<tr>
<td>+=</td>
</tr>
<tr>
<td>-=</td>
</tr>
<tr>
<td>*=</td>
</tr>
<tr>
<td>/=</td>
</tr>
<tr>
<td>%=</td>
</tr>
</tbody>
</table>

#### Assignment Operators

<table>
<thead>
<tr>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>!=</td>
</tr>
</tbody>
</table>

9/22/08
Practice

What is the value of the following Relational Expressions?

```java
int width = 99, height = 42;
width > height
-width <= height
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!

```c
if ( condition )
{
    //statements
}
```
int age;
const int VOTING_AGE = 18;

cin >> age;
if ( age >= VOTING_AGE )
{
    cout << age << " > " << VOTING_AGE;
    Cout << " You can vote!"
    << endl;
}
Practice

• 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?
Practice

```cpp
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?

// Value is: " << value;
```
Practice

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

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

• Where you born before 1990?

• Is you age evenly divisible by 7?
Coding Standards

```cpp
if( expression )
{
    statement 1;
}
```

If you only have ONE statement in the body of the if, the `{ }` are optional in C++.

For this class, the `{ }` must **ALWAYS** be used. Not using `{ }` will result in a loss of style points.

The `{ }` must also be on their own line.

Why?
More on Truth

int x = 5, y = 0;
if ( y - x )
{
    // This will be executed
    cout << "y - x is True" << endl;
}
if ( y )
{
    // This will NOT be executed
    cout << "y is True" << endl;
}