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

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

```c
int x=9, y=42;
x > y
y == x    // y = x; is the assignment operator
x <= (x * y + 99)
```
## 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

9/22/08
Precedence (page 1101)

<table>
<thead>
<tr>
<th>Precedence Operators</th>
<th>(Highest to Lowest)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(unary negation)</td>
<td>-</td>
</tr>
<tr>
<td>*</td>
<td>/</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;=</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;=</td>
</tr>
<tr>
<td>==</td>
<td>!=</td>
</tr>
<tr>
<td>=</td>
<td>+=</td>
</tr>
<tr>
<td>-=</td>
<td>*=</td>
</tr>
<tr>
<td>/=</td>
<td>%=</td>
</tr>
</tbody>
</table>

Arithmetic Operators

Relational Operators

Assignment Operators

9/22/08

CS150 Introduction to Computer Science 1
Practice

What is the value of the following Relational Expressions?

```plaintext
int x = 99, y = 42;
x > y
y <= x
y != x
x == (x + 1)
y == y + 1
y == x - 45
```

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!
Practice: What is the output?

```cpp
int x=5, y=10;

if ( x < y )
{
    cout << x << " < " << y;
    cout << " is true" << endl;
}
```
Practice

• For the problem below:
  o what data will you need?
  o 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 showing the letter grade the student received and a message stating if the student passed the course.
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 = ??

// what does this output look like?

cout << "Value is: " << value;
Practice

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

```cpp
bool value;
int yourAge = 22, currentYear = 2008;
```

• Are you old enough to vote?

• Where you born before 1980?

• 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

- Expressions that evaluate to non-zero are considered **true**

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