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)

```
int x=9, y=42;

x > y

y == x // y = x; is the assignment operator

x <= (x * y + 99)
```

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)

	Precedence Operators (Highest to Lowest)
	(unary negation) -
Arithmetic Operators	* / %
	+ -
Relational Operators	> >= < <=
	== !=
Assignment Operators	= += -= *= /= %=

Practice

- What is the value of the following Relational Expressions?

```
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?

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


if ( x < y )
{
   cout << x << “ < ” << y;
   cout << “ is true” << 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 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?

Practice

```
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?

```
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

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

~~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**

```
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;
}
```