INPUT, LOGICAL OPERATIONS, CONTROL STRUCTURES



• We are already familar with output:

```
int x;
x = 5;
cout << ''The value of x is '' << x << endl;</pre>
```

• C++ makes simple input just as straightforward:

```
int x;
cout << ''Enter a value for x: '';
cin >> x;
cout << ''The value of x is '' << x << endl;</pre>
```

(We'll look at more complex input later).

Today • Input with cin statement • The if statement • Relational operators Logical operators • Truth tables • The if-else statement. • The while statement cis1.5-fall2009-parsons-lectII.1

- The function that reads input is the cin.
- Note the use of the >>, which tells cin which variable to read the value into.
- The message:

cout << ''Enter a value for x: '';</pre>

is called a *prompt*.

• A prompt tells the user what to do.

cis1.5-fall2009-parsons-lectII.1

3

cis1.5-fall2009-parsons-lectII.1

Why we need control structures

- So far, all we have seen is how to make a program do a *sequence* of things.
 - -goNorth()
 - -goEast()
 - -goNorth()
- There is more to life than this!
- We want to be able to make a program:
 - Choose between doing different things
 - Do the same thing several times
- C++ gives us *control structures* which allow us to do these things.

cis1.5-fall2009-parsons-lectII.1

The if branching statement

- The if is a *conditional*
 - Means the computer makes a choice
- It is also a *control structure*
- General structure:

```
if(<something that is true or false>)
{
```

<some instructions>

- }
- cis1.5-fall2009-parsons-lectII.1

The if branching statement

- Perhaps the simplest control structure is the if statement.
- Consider the robot from the homework.
- To tell when the robot is in the middle of the grid, we need to do:
- if(x == 4) {
 cout << "The robot is in the middle";
 }</pre>
- Let's look at it in a bit more detail.

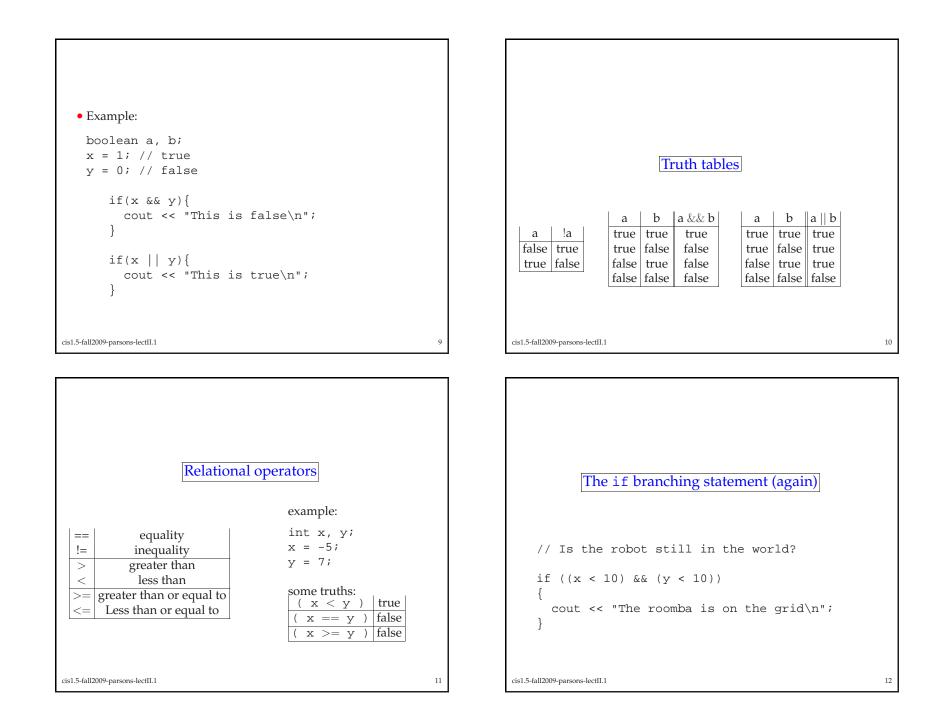
```
cis1.5-fall2009-parsons-lectII.1
```

Boolean expressions

- Boolean expressions are things that are true or false.
- Boolean variables: true (1) or false (0)
- Logical operators:

!	not
&&	and
	or

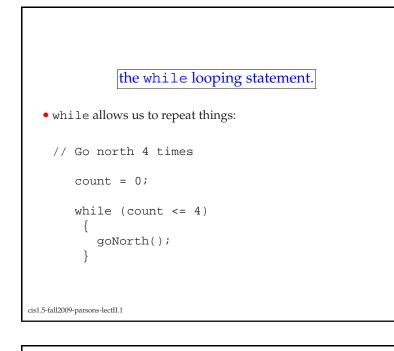
cis1.5-fall2009-parsons-lectII.1





cis1.5-fall2009-parsons-lectII.1

16



• General structure:	
while(<something false="" is="" or="" that="" true="">) {</something>	
<some instructions=""></some>	
}	
• This structure looks a lot like if	

18

cis1.5-fall2009-parsons-lectII.1

17

19

Summary

- We talked about simple input using cin.
- We covered some of the basic control structures:

-if,while

- Along the way we looked at boolean expressions and relational operators as well.
- Now it is time to read Chapter 2 of the textbook.

cis1.5-fall2009-parsons-lectII.1