Advantages of functions

• Modularity

- We can divide up a program into small, understandable pieces (kind of like steps in a recipe)
- This makes the program easier to read
- This makes the program easier to *debug*.

• Write once, use many times

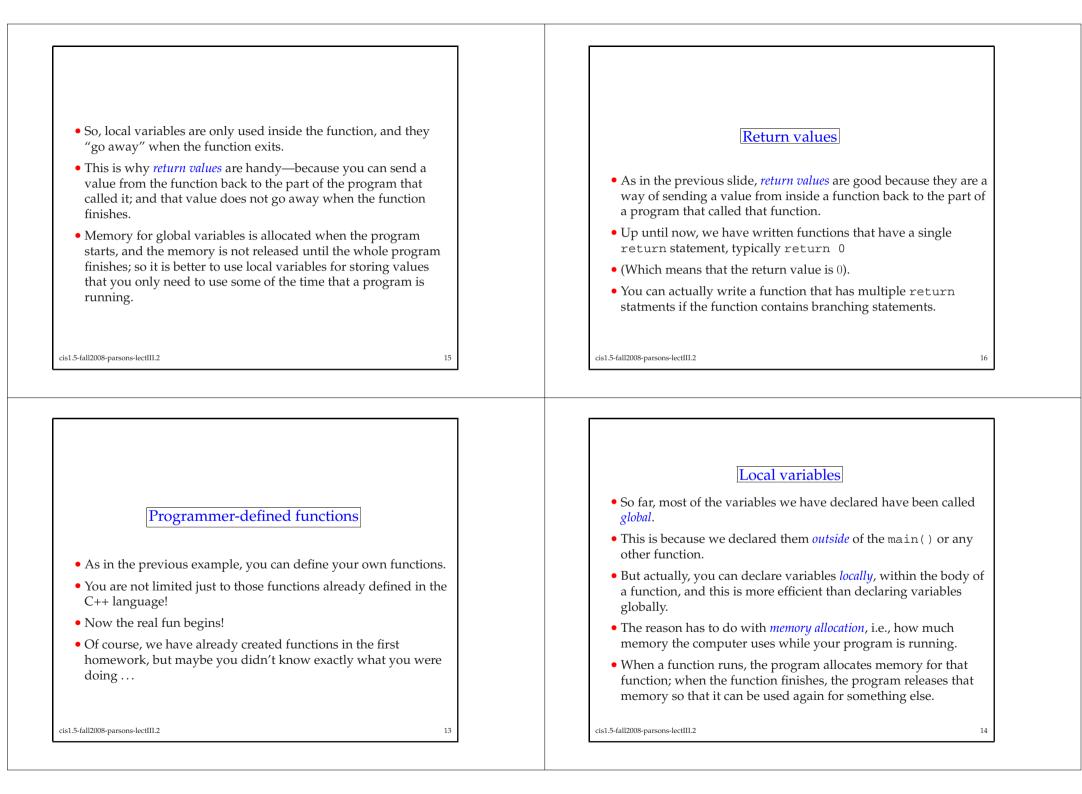
- If we have a task that will be performed many times, we only have to *define* a function once; then we can *call* (or *invoke*) the function as many times as we need it
- Also, we can use *parameters* (or *arguments*) to use the function to perform the same task on or with different data values

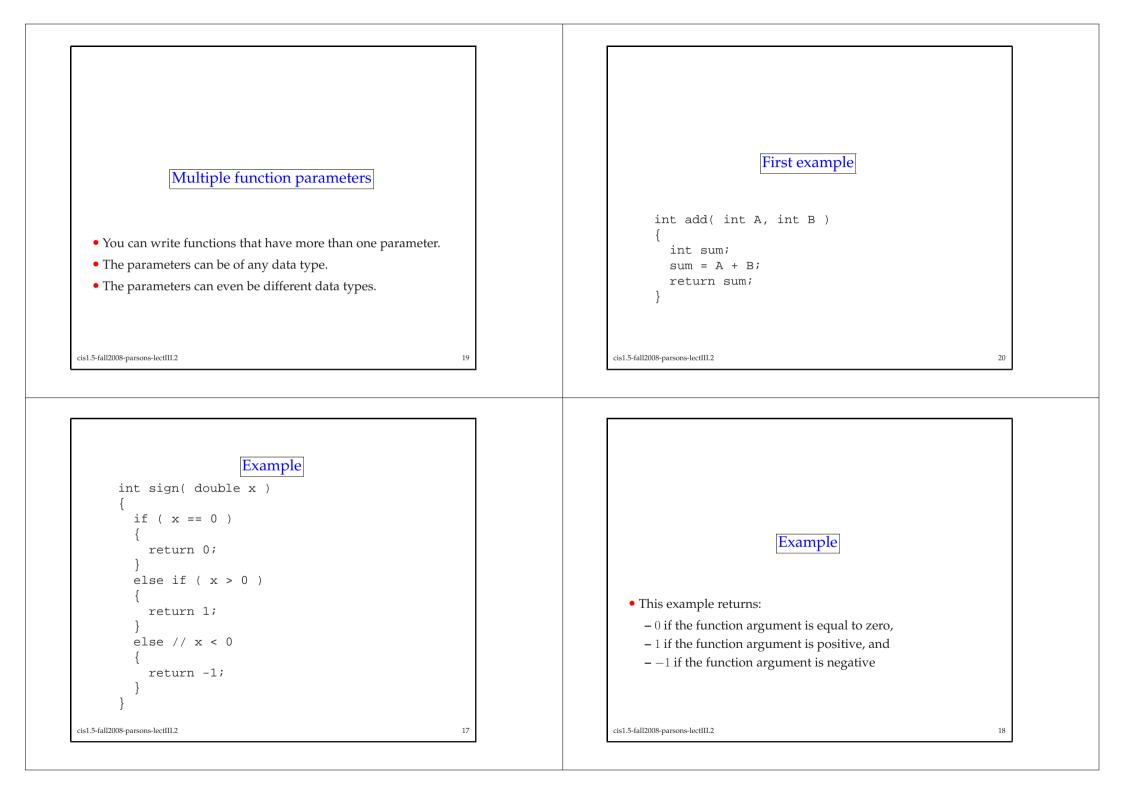
cis1.5-fall2008-parsons-lectIII.2

Library functions • We have already talked about *built-in*, or *library*, functions • These are functions that come with the C++ language • We have used the *iostream* C++ library: • iostream.cout • iostream.cin • We have used the *fstream* C++ library: • ifstream.open • ifstream.close • ofstream.open • ofstream.close

FUNCTIONS	Today • What are functions and why to use them – I will use the terms "function" and "method" interchangeably. • Library and programmer-defined functions • Parameters and return values • Reading: textbook Chapter 4
	cis1.5-fall2008-parsons-lectIII.2 2

A first example	A second example
#include <iostream></iostream>	#include <iostream></iostream>
using namespace std;	using namespace std;
<pre>void sayHello() // define function</pre>	<pre>void sayHello(); // function header only</pre>
{	int main()
<pre>cout << "hello\n";</pre>	{
return 0;	<pre>sayHello(); // call function </pre>
}	return 0;
int main()	
{	<pre>void sayHello() // define function</pre>
sayHello(); // call function	<pre>{ cout << "hello\n";</pre>
return 0;	return 0;
}	
J	}
	}
J 5-fall2008-parsons-lectIII.2	7 cis1.5-fall2008-parsons-lectIII.2 8
J 5-fall2008-parsons-lectIII.2	
• We have also mentioned the <i>math</i> C library:	7 cis1.5-fall2008-parsons-lectIII.2 8
	How functions work
• We have also mentioned the <i>math</i> C library: - sqrt - pow	
• We have also mentioned the <i>math</i> C library: – sqrt	How functions work • Functions must be <i>defined</i> (or "declared") and then they can be





Second example

21

```
int doMath( int A, int B, char op )
{
    int result;
    if ( op=='+' )
    {
        result = A + B;
    }
    else if ( op=='-' )
    {
        result = A - B;
    }
    else if ( op=='*' )
    {
        result = A * B;
    }
        return result;
    } // end of doMath()

cisl.5-fall2008-parsons-lectIII.2
```

Summary Nis lecture has introduced functions. We briefly recapped library functions. Then considered user-defined functions. In our consideration of user-defined fuctions, we looked at: Parameters. Local variables Return values