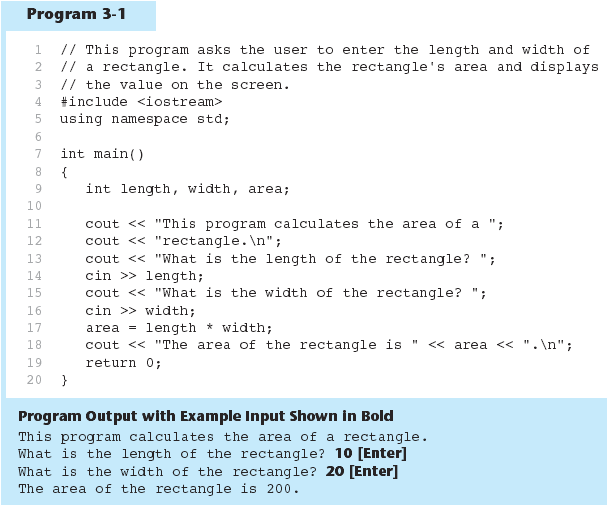
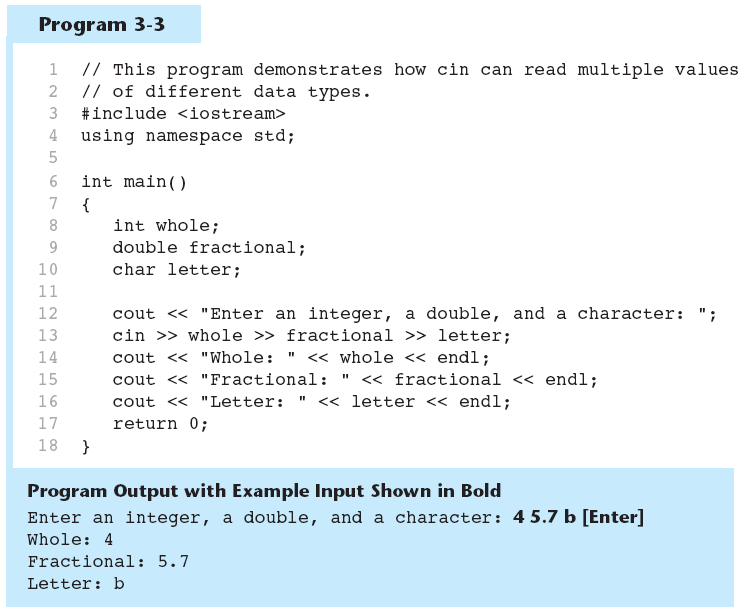
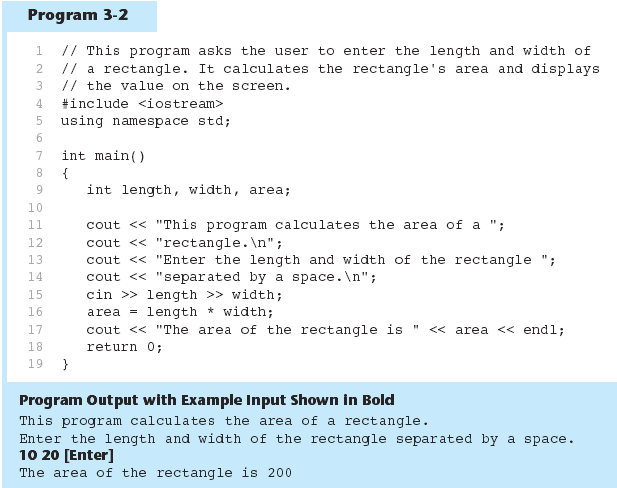
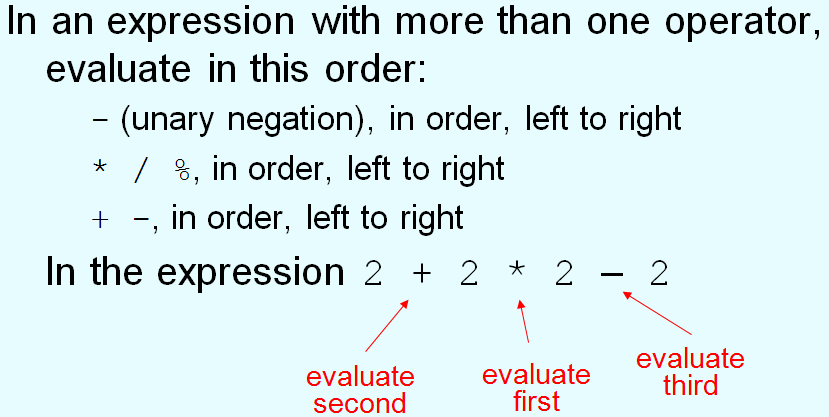
**Chapter 3: Expressions and Interactivity**

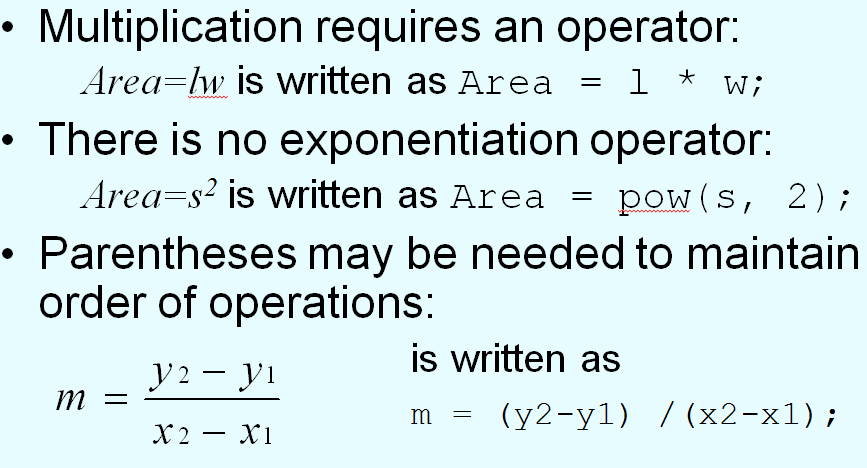
The cin object can be used to read data typed at the keyboard.

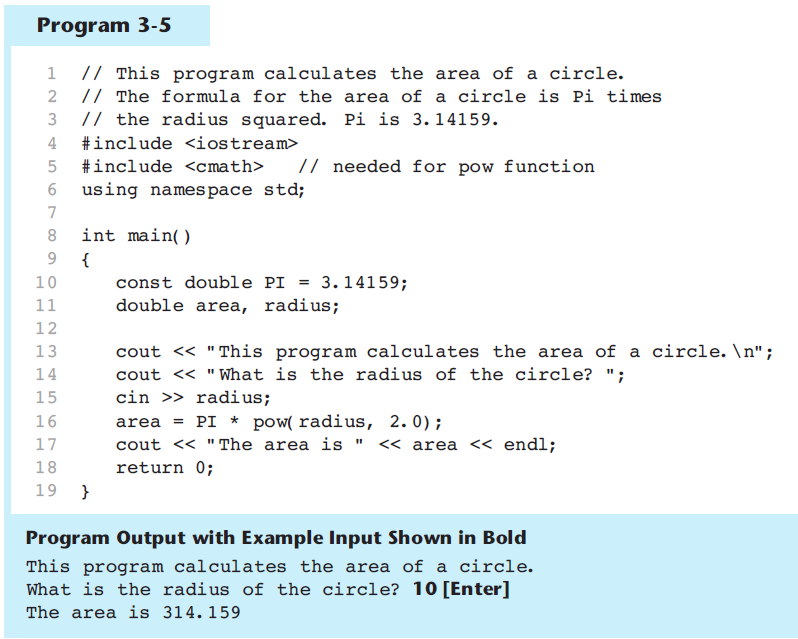
Just as cout is C++ s standard output object, Cin is the standard input object.









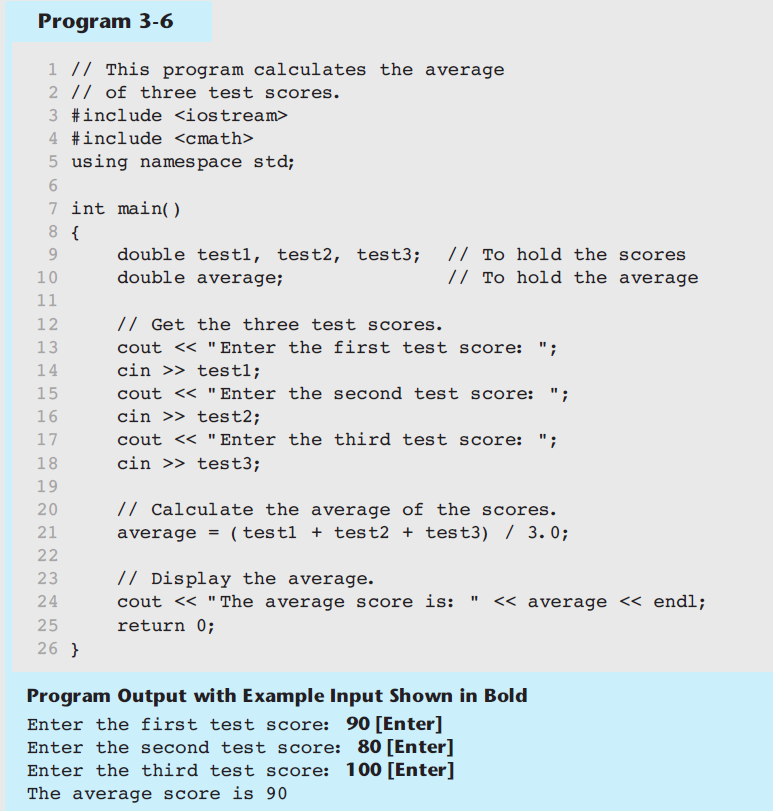


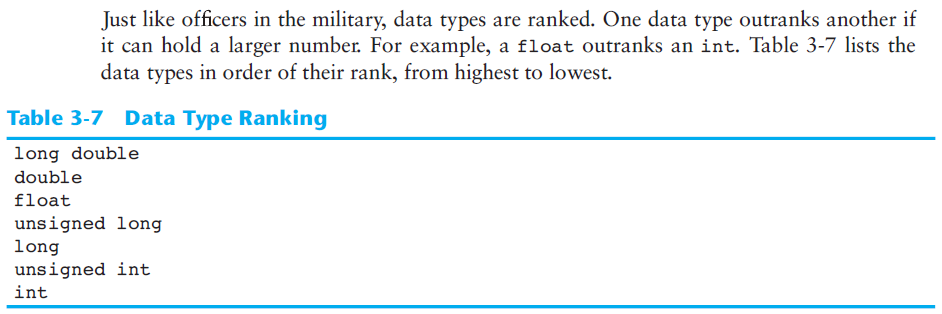
**NOTE:** Program 3-5 is presented as a demonstration of the pow function. In reality, there is no reason to use the pow function in such a simple operation. The math statement could just as easily be written as

area = PI \* radius \* radius; The pow function is useful, however, in operations that involve larger exponents.

The statement area = pow (4.0, 2.0) is equivalent to the following algebraic statement: area = 42

Here is another example of a statement using the pow function. It assigns 3 times 63 to x: x = 3 \* pow(6.0, 3.0);





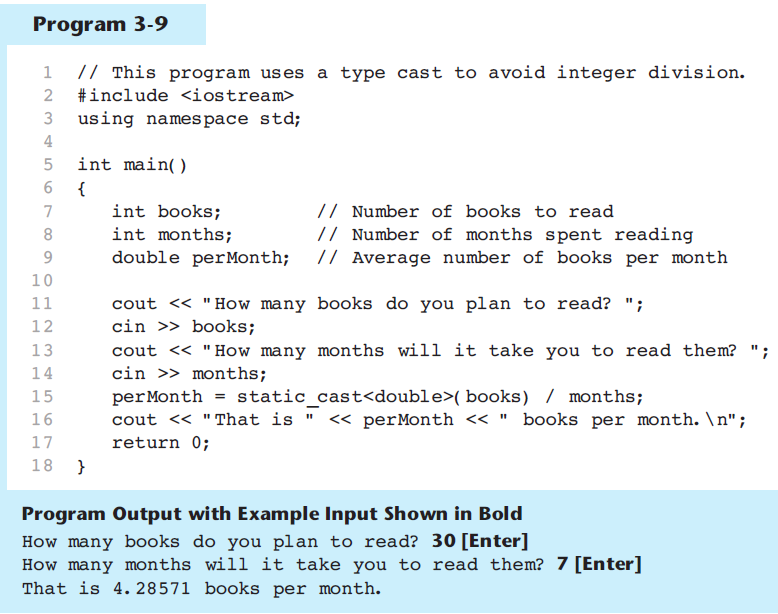
This automatic conversion is known as type coercion. When a value is converted to a higher data type, it is said to be promoted. To demote a value means to convert it to a lower data type.

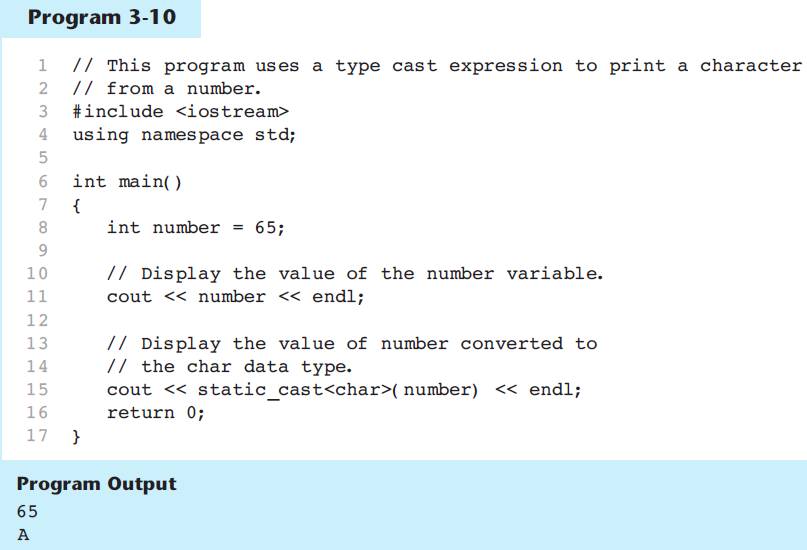
Coercion Rules:

1) char, short, unsigned short automatically promoted to int

2) When operating on values of different data types, the lower one is promoted to the type of the higher one.

3) When using the = operator, the type of expression on right will be converted to type of variable on left





C++ also supports two older methods of creating type cast expressions: the C-style form and the prestandard C++ form. The C-style cast is the name of a data type enclosed in parentheses, preceding the value that is to be converted. For example, the following statement converts the value in number to an int.

val = (int)number;

The following statement shows another example.

perMonth = (double)books / months;

In this statement the value in the books variable is converted to a double before the division takes place.

The prestandard C++ form of the type cast expression appears as a data type name followed by a value inside a set of parentheses. Here is an example:

val = int(number);

The type cast in this statement returns a copy of the value in number, converted to an int.

* C-Style cast: data type name in ()

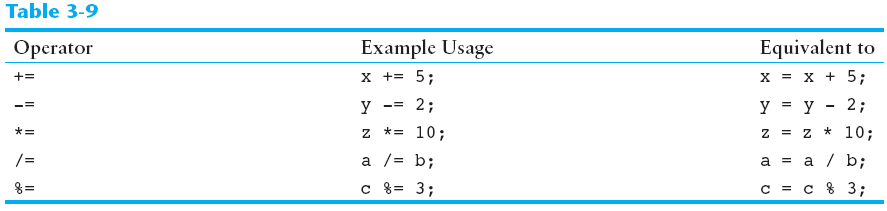
cout << ch << " is " << (int)ch;

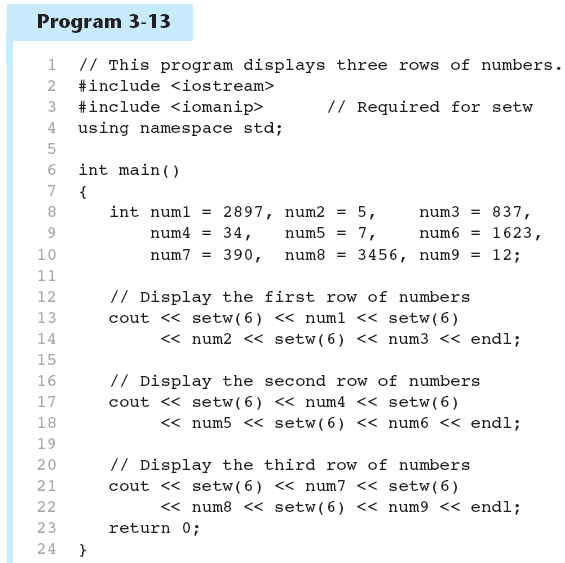
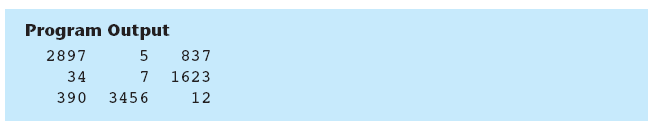
* Prestandard C++ cast: value in ()

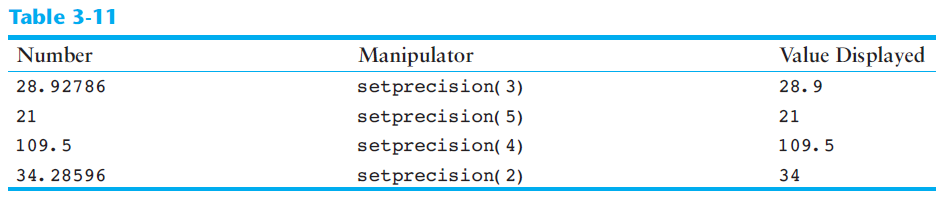
cout << ch << " is " << int(ch);

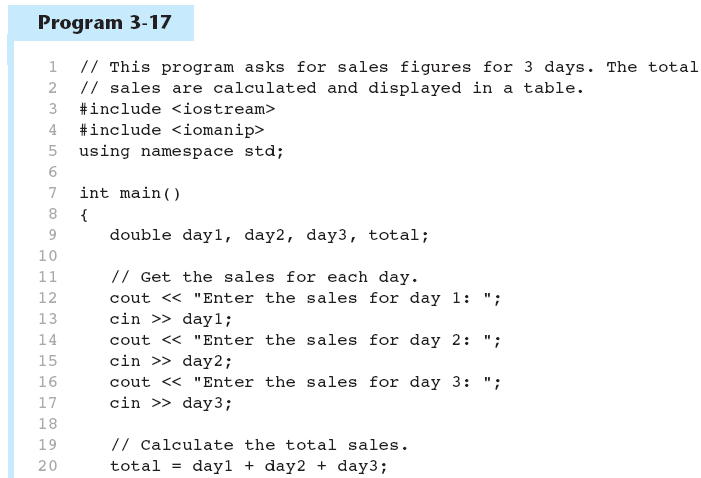
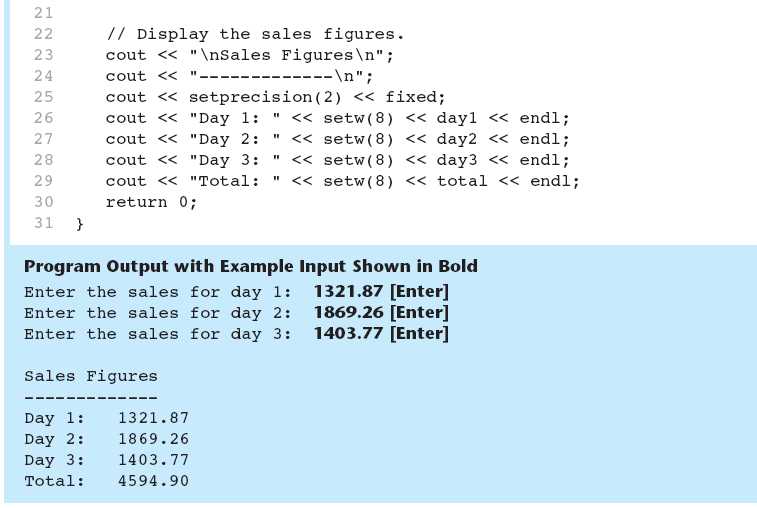
* Both are still supported in C++, although static\_cast is preferred
* The combined assignment operators provide a shorthand for these types of statements.
* The statement: sum = sum + 1;

is equivalent to sum += 1;

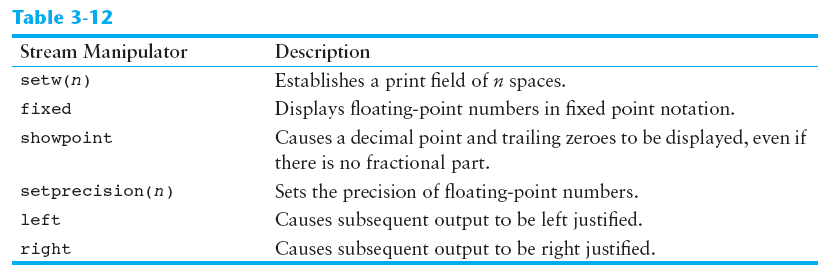


  **NOTE:** A new header le, iomanip, is included in Program 3-13. It must be used in any program that uses setw.

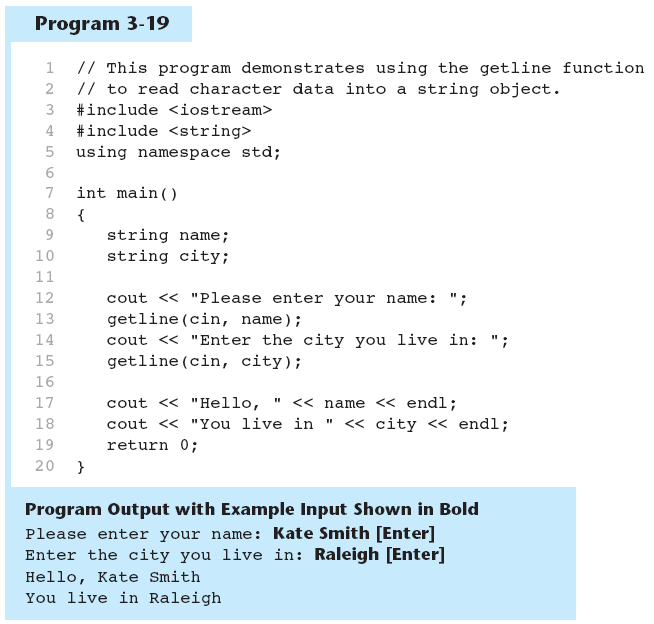


* Some affect values until changed again:
  + fixed: use decimal notation for floating-point values
  + setprecision(x): when used with fixed, print floating-point value using x digits after the decimal. Without fixed, print floating-point value using x significant digits
  + showpoint: always print decimal for floating-point values



Working with Characters and **string** Objects: **getline**



* To read a single character:
  + Use cin:

char ch;

cout << "Strike any key to continue";

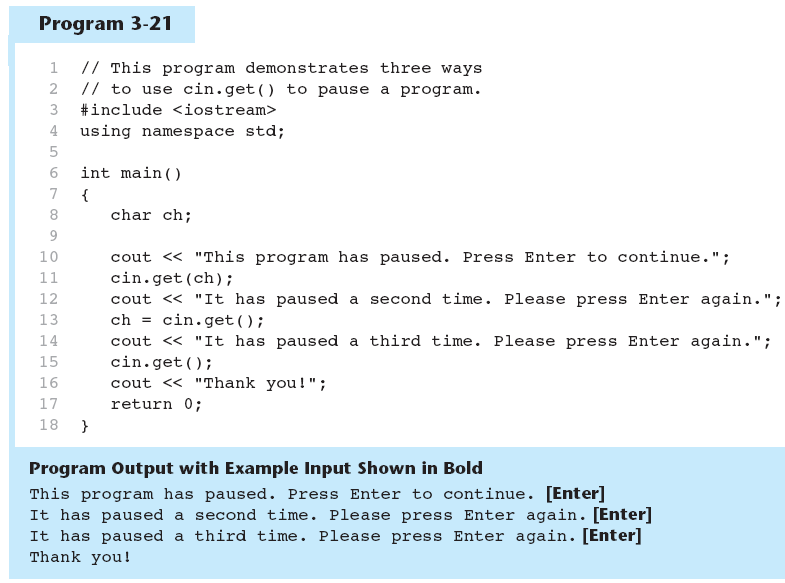
cin >> ch;

Problem: will skip over blanks, tabs, <CR>

* + Use cin.get():

cin.get(ch);

Will read the next character entered, even whitespace



* Mixing cin >> and cin.get() in the same program can cause input errors that are hard to detect
* To skip over unneeded characters that are still in the keyboard buffer, use cin.ignore():

cin.ignore(); // skip next char

cin.ignore(10, '\n'); // skip the next

// 10 char. or until a '\n'

* To find the length of a string:

string state = "Texas";

int size = state.length();

* To concatenate (join) multiple strings:

string greeting1 = "Hello ";

string greeting2;

string name1 = "World";

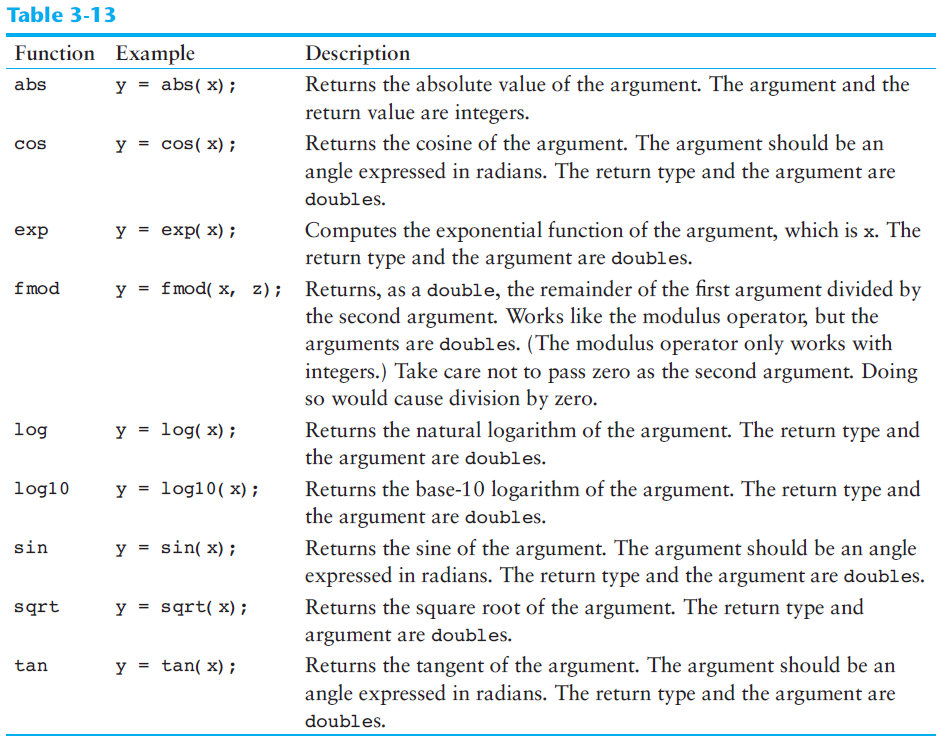
string name2 = "People";

greeting2 = greeting1 + name1; // greeting2 now holds "Hello World"

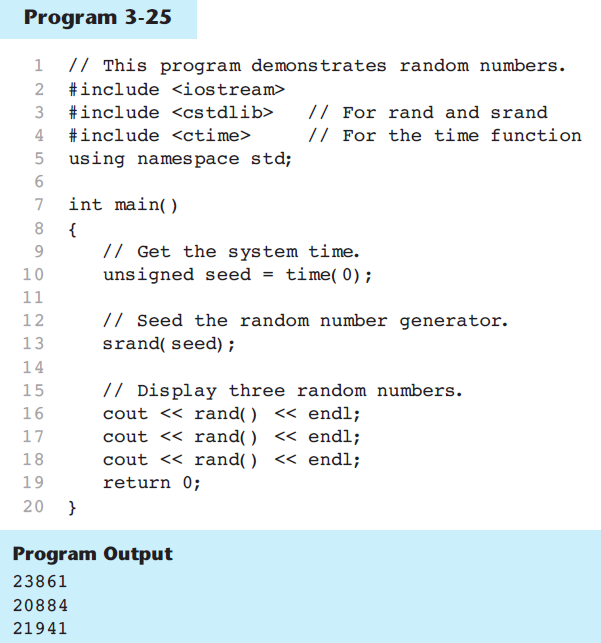
greeting1 = greeting1 + name2; // greeting1 now holds "Hello People"

Or using the **+=** combined assignment operator:

greeting1 += name2;



* These require cstdlib header file
* rand(): returns a random number (int) between 0 and the largest int the compute holds. Yields same sequence of numbers each time program is run.
* srand(x): initializes random number generator with unsigned int x



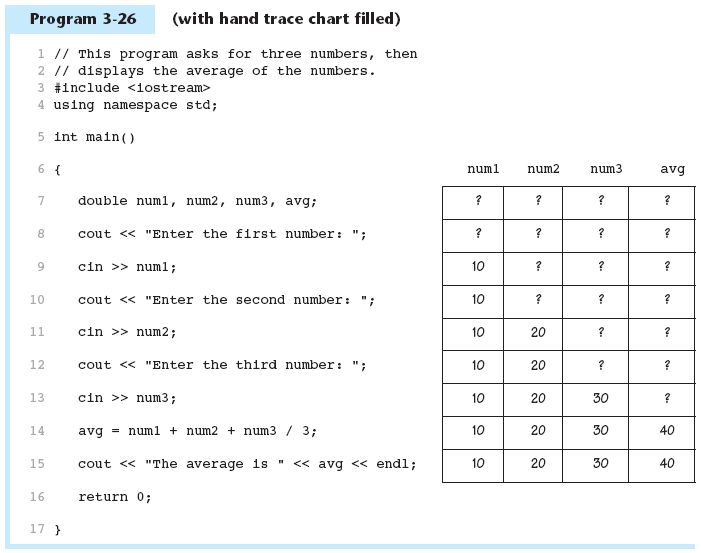
**NOTE:** If you wish to limit the range of the random number, use the following formula.

y = 1 + rand() % maxRange;

The maxRange value is the upper limit of the range. For example, if you wish to generate

a random number in the range of 1 through 100, use the following statement.

y = 1 + rand() % 100;



A Case Study

Psuedocode:

*Ask the user to input the crate's length.*

*Ask the user to input the crate's width.*

*Ask the user to input the crate's height.*

*Calculate the crate's volume.*

*Calculate the cost of building the crate.*

*Calculate the customer's charge for the crate.*

*Calculate the profit made from the crate.*

*Display the crate's volume.*

*Display the cost of building the crate.*

*Display the customer's charge for the crate.*

*Display the profit made from the crate.*

