`string` Member Functions and Operators

- To find the length of a string:

  ```
  string state = "Texas";
  int size = state.length();
  ```

- To concatenate (join) multiple strings:

  ```
  greeting2 = greeting1 + name1;
  greeting1 = greeting1 + name2;
  ```

  Or using the += combined assignment operator:

  ```
  greeting1 += name2;
  ```

- stringDemo.cpp

3.9

More Mathematical Library Functions

More Mathematical Library Functions

- Require `cmath` header file 3.24
- Take `double` as input, return a `double`
- Commonly used functions:

`sin`	Sine
`cos`	Cosine
`tan`	Tangent
`sqrt`	Square root
`log`	Natural (e) log
`abs`	Absolute value (takes and returns an int)

- mathsFunctions.cpp

- Go to Random numbers chapter.

More Mathematical Library Functions

- 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`
- 3.25

Generate Random numbers within a range

- Get one end of the range and store -> end1
- Get the other end and store -> end2
- Range = end2 – end1 + 1
- Seed the random number generator with current time
- Output = end1 + random() % range
- `Rand_test1.cpp`

3.10

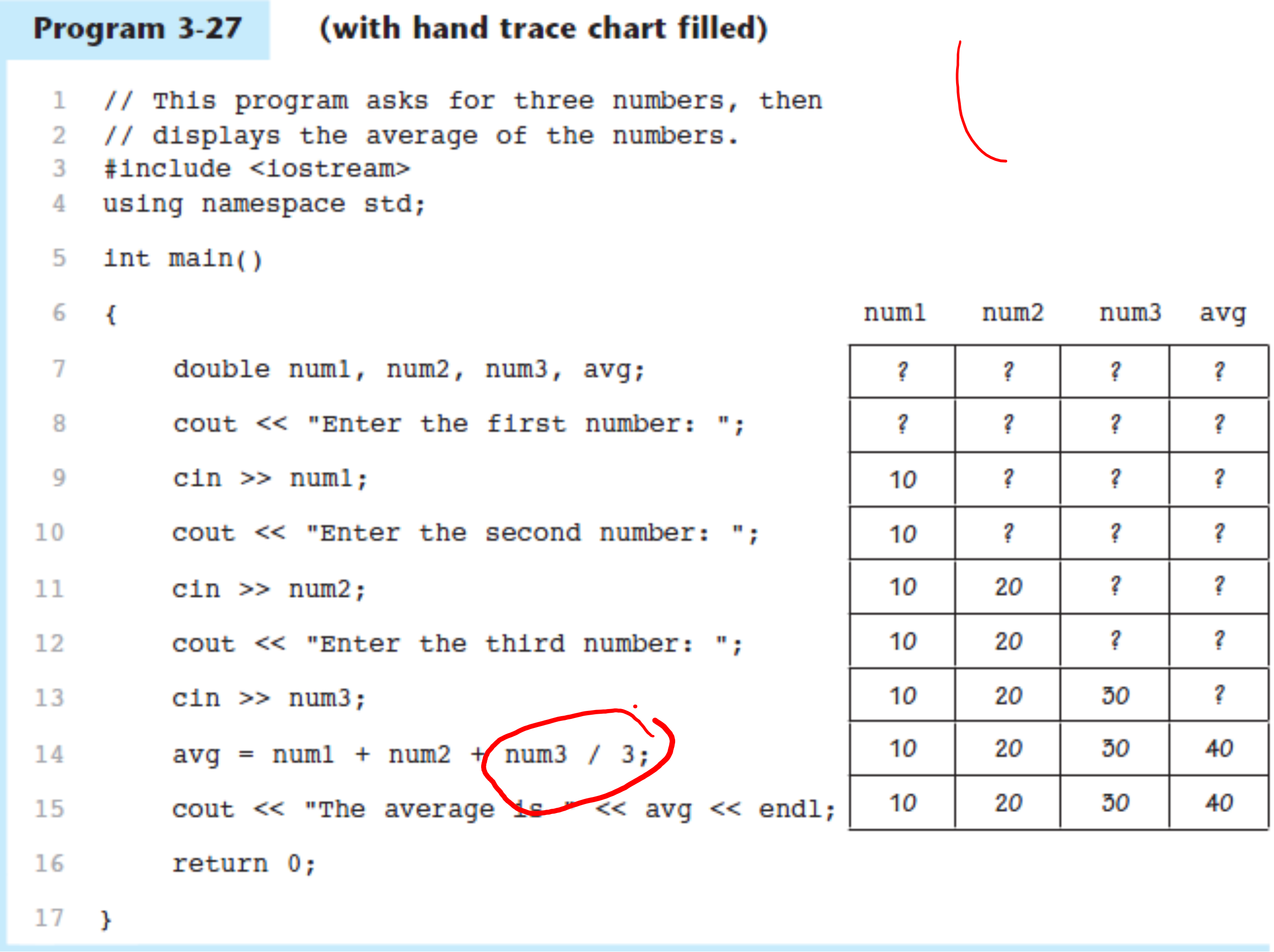
Hand Tracing a Program

Hand Tracing a Program

- Hand trace a program: act as if you are the computer, executing a program:
  - step through and 'execute' each statement, one-by-one
  - record the contents of variables after statement execution, using a hand trace chart (table)

- Useful to locate logic or mathematical errors

Program 3-27 with Hand Trace Chart



Program 3-27 **(with hand trace chart filled)**

```
// This program asks for three numbers, then
// displays the average of the numbers.
#include <iostream>
using namespace std;

int main()
{
    double num1, num2, num3, avg;
    cout << "Enter the first number: ";
    cin >> num1;
    cout << "Enter the second number: ";
    cin >> num2;
    cout << "Enter the third number: ";
    cin >> num3;
    avg = num1 + num2 + num3 / 3;
    cout << "The average is " << avg << endl;
    return 0;
}
```

num1	num2	num3	avg
?	?	?	?
?	?	?	?
10	?	?	?
10	?	?	?
10	20	?	?
10	20	?	?
10	20	30	?
10	20	30	40
10	20	30	40

3.11

A Case Study

A Case Study

- General Crates, Inc. builds custom-designed wooden crates.
- You have been asked to write a program that calculates the:
  - Volume (in cubic feet)
  - Cost
  - Customer price
  - Profit of any crate GCI builds

Variables

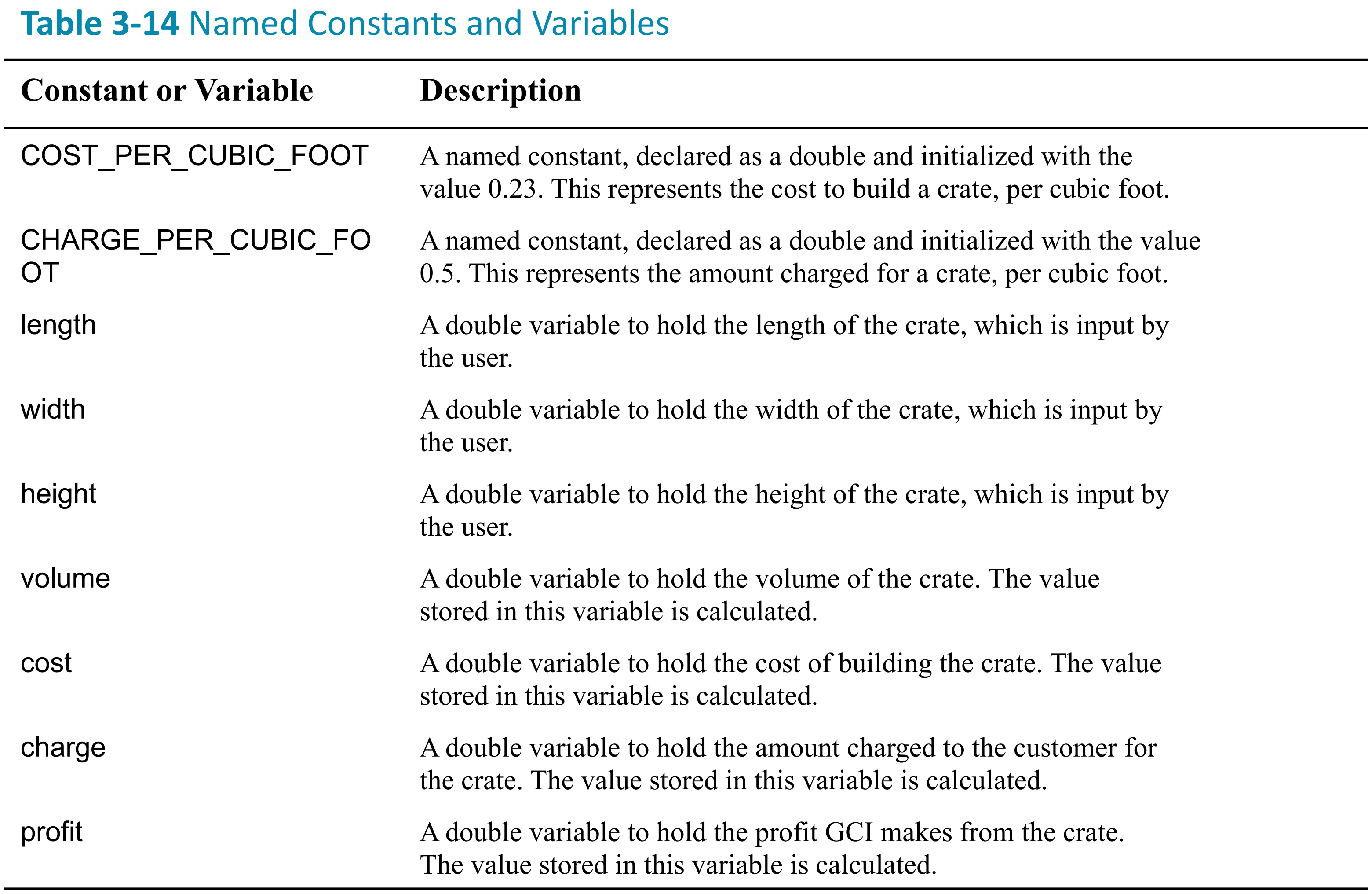


Table 3-14 Named Constants and Variables

Constant or Variable	Description
COST_PER_CUBIC_FOOT	A named constant, declared as a double and initialized with the value 0.23. This represents the cost to build a crate, per cubic foot.
CHARGE_PER_CUBIC_FOOT	A named constant, declared as a double and initialized with the value 0.5. This represents the amount charged for a crate, per cubic foot.
length	A double variable to hold the length of the crate, which is input by the user.
width	A double variable to hold the width of the crate, which is input by the user.
height	A double variable to hold the height of the crate, which is input by the user.
volume	A double variable to hold the volume of the crate. The value stored in this variable is calculated.
cost	A double variable to hold the cost of building the crate. The value stored in this variable is calculated.
charge	A double variable to hold the amount charged to the customer for the crate. The value stored in this variable is calculated.
profit	A double variable to hold the profit GCI makes from the crate. The value stored in this variable is calculated.

Program Design

The program must perform the following general steps:

Step 1:

- Ask the user to enter the dimensions of the crate

Step 2:

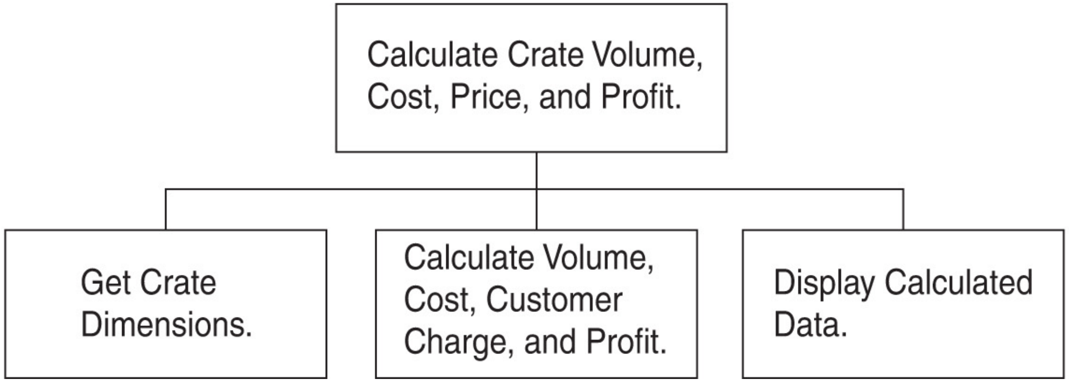
- Calculate:
  - the crate’s volume
  - the cost of building the crate
  - the customer’s charge
  - the profit made

Step 3:

- Display the data calculated in Step 2.

General Hierarchy Chart

Figure 3-7



Calculate Crate Volume, Cost, Price, and Profit.

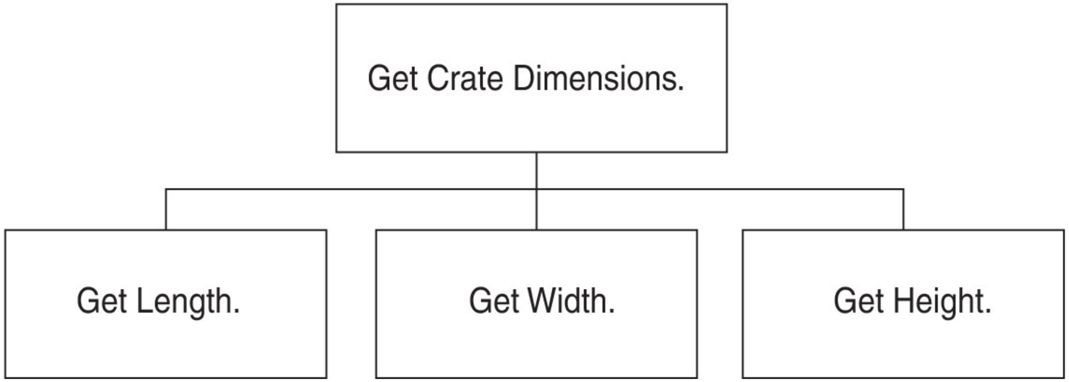
Get Crate Dimensions.

Calculate Volume, Cost, Customer Charge, and Profit.

Display Calculated Data.

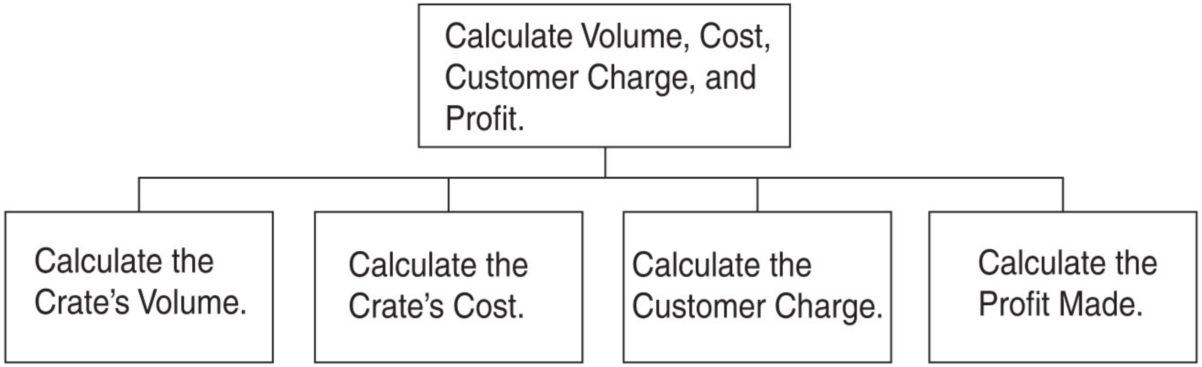
Get Crate Dimensions

Figure 3-8



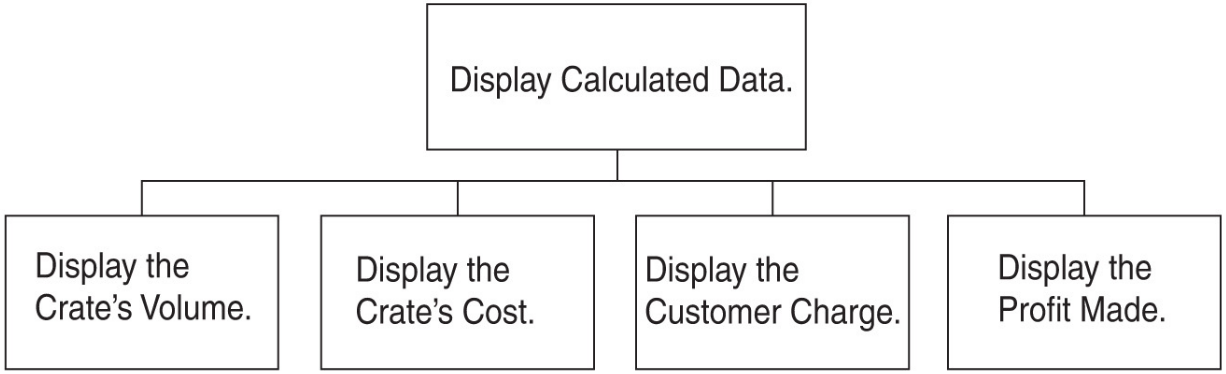
Calculate Volume, Cost, Customer Charge, and Profit

Figure 3-9


Calculate Volume, Cost, Customer Charge, and Profit.
Calculate the Crate's Volume.
Calculate the Crate's Cost.
Calculate the Customer Charge.
Calculate the Profit Made.

Display Calculated Data

Figure 3-10


Display Calculated Data.

Display the Crate's Volume.

Display the Crate's Cost.

Display the Customer Charge.

Display the Profit Made.

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.
```

Calculations

The following formulas will be used to calculate the crate’s volume, cost, charge, and profit:

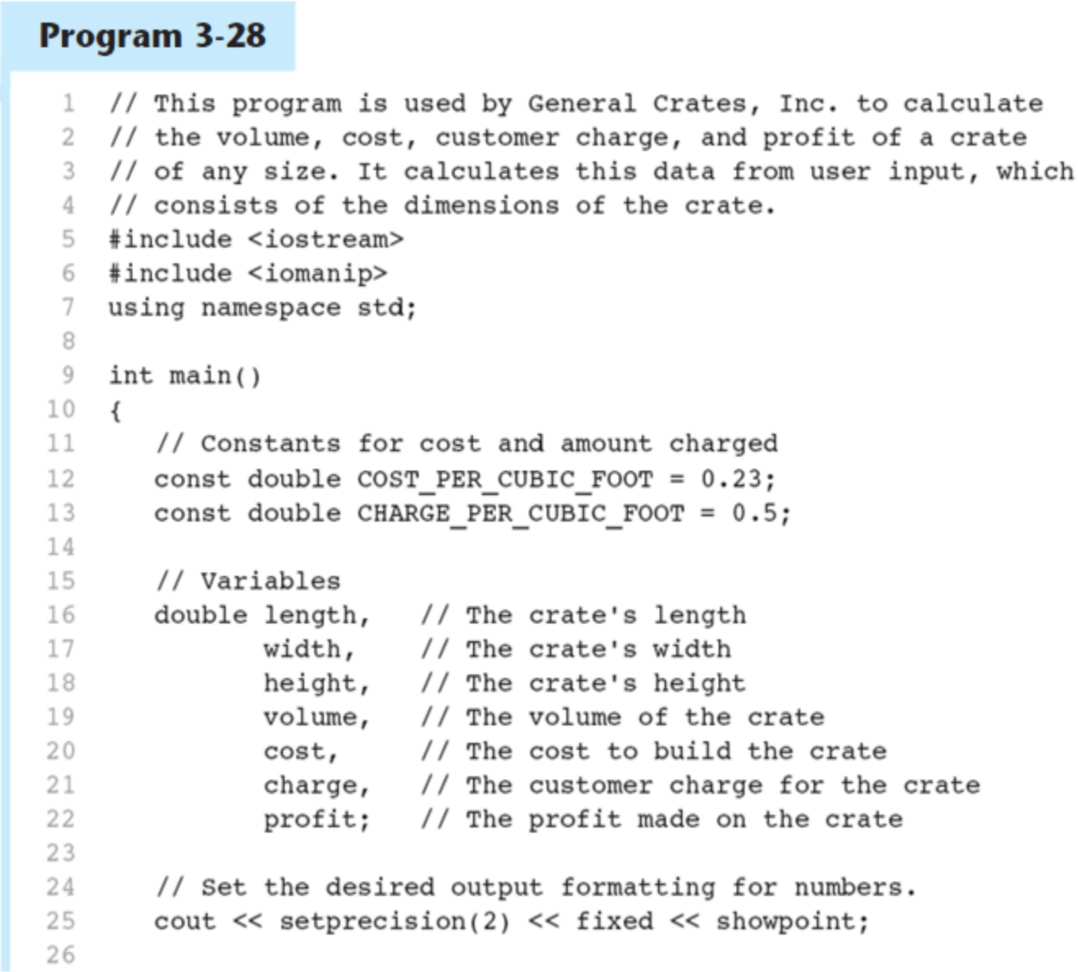
volume = length × width × height

cost = volume × 0.23

charge = volume × 0.5

profit = charge − cost

The Program

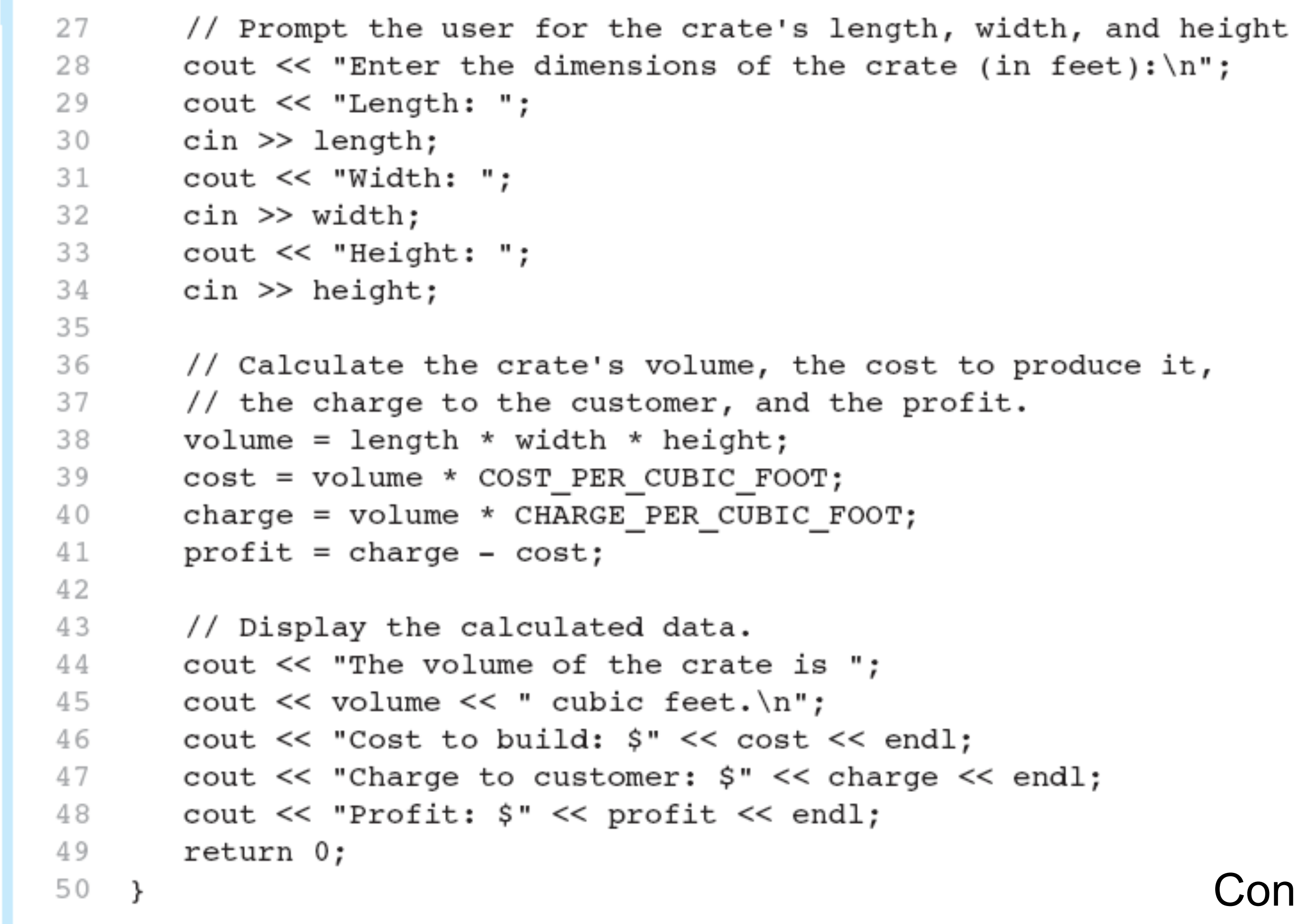


Program 3-28

```
// This program is used by General Crates, Inc. to calculate
// the volume, cost, customer charge, and profit of a crate
// of any size. It calculates this data from user input, which
// consists of the dimensions of the crate.
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
   // Constants for cost and amount charged
   const double COST_PER_CUBIC_FOOT = 0.23;
   const double CHARGE_PER_CUBIC_FOOT = 0.5;

   // Variables
   double length,   // The crate's length
          width,    // The crate's width
          height,   // The crate's height
          volume,   // The volume of the crate
          cost,     // The cost to build the crate
          charge,   // The customer charge for the crate
          profit;   // The profit made on the crate

   // Set the desired output formatting for numbers.
   cout << setprecision(2) << fixed << showpoint;

```

Continued…

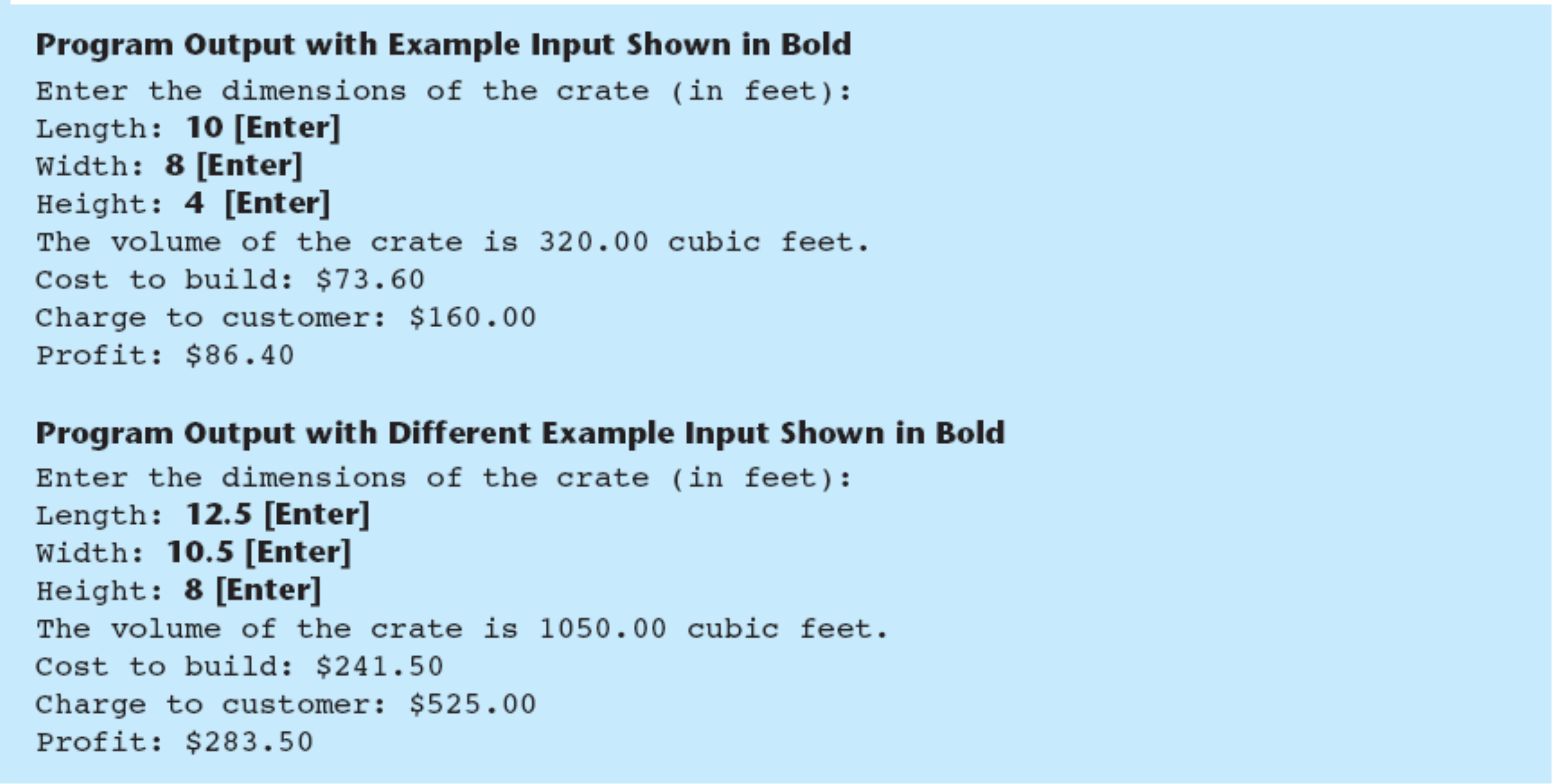
The Program



```
    // Prompt the user for the crate's length, width, and height
    cout << "Enter the dimensions of the crate (in feet):\n";
    cout << "Length: ";
    cin >> length;
    cout << "Width: ";
    cin >> width;
    cout << "Height: ";
    cin >> height;

    // Calculate the crate's volume, the cost to produce it,
    // the charge to the customer, and the profit.
    volume = length * width * height;
    cost = volume * COST_PER_CUBIC_FOOT;
    charge = volume * CHARGE_PER_CUBIC_FOOT;
    profit = charge - cost;

    // Display the calculated data.
    cout << "The volume of the crate is ";
    cout << volume << " cubic feet.\n";
    cout << "Cost to build: $" << cost << endl;
    cout << "Charge to customer: $" << charge << endl;
    cout << "Profit: $" << profit << endl;
    return 0;
}
```

Continued…

The Program



Program Output with Example Input Shown in Bold

```
Enter the dimensions of the crate (in feet):
Length: 10 [Enter]
Width: 8 [Enter]
Height: 4  [Enter]
The volume of the crate is 320.00 cubic feet.
Cost to build: $73.60
Charge to customer: $160.00
Profit: $86.40
```

Program Output with Different Example Input Shown in Bold

```
Enter the dimensions of the crate (in feet):
Length: 12.5 [Enter]
Width: 10.5 [Enter]
Height: 8 [Enter]
The volume of the crate is 1050.00 cubic feet.
Cost to build: $241.50
Charge to customer: $525.00
Profit: $283.50
```