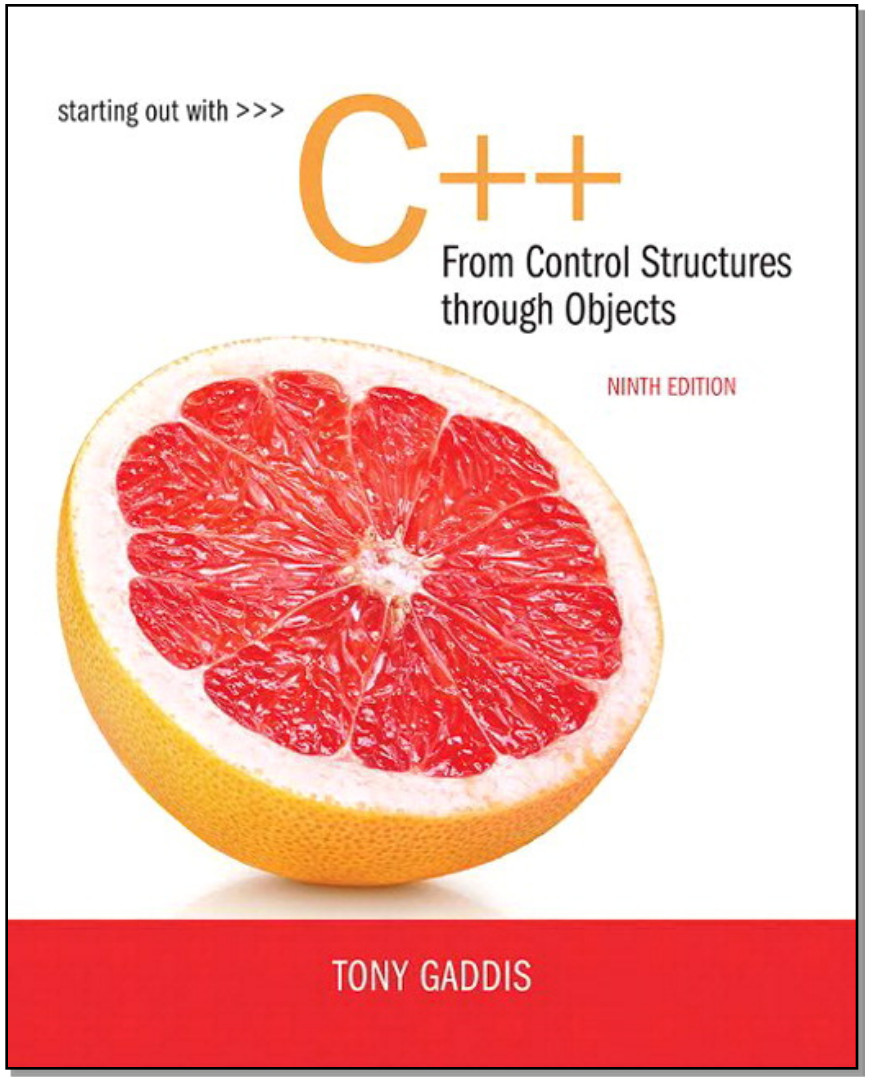
STARTING OUT WITH C++

9th Edition



Chapter 3

التعابير والتفاعل

Expressions and Interactivity

3.1

cin الكائن

The cin Object

The `cin` Object

- كائن الإدخال المعياري
- يتطلب ملف iostream مثل cout
- يستخدم لقراءة الدخل من لوحة المفاتيح
- تستعاد المعلومات من cin بواسطة >>
- يخزن الدخل في متغير أو أكثر

The `cin` Object in Program 3-1

Program 3-1

```
1  // This program asks the user to enter the length and width of
2  // a rectangle. It calculates the rectangle's area and displays
3  // the value on the screen.
4  #include <iostream>
5  using namespace std;
6
7  int main()
8  {
9      int length, width, area;
10
11     cout << "This program calculates the area of a ";
12     cout << "rectangle.\n";
13     cout << "What is the length of the rectangle? ";
14     cin >> length;
15     cout << "What is the width of the rectangle? ";
16     cin >> width;
17     area = length * width;
18     cout << "The area of the rectangle is " << area << ".\n";
19     return 0;
20 }
```

Program Output with Example Input Shown in Bold

```
This program calculates the area of a rectangle.
What is the length of the rectangle? 10 Enter
What is the width of the rectangle? 20 Enter
The area of the rectangle is 200.
```

The `cin` Object

- يحول **`cin`** البيانات إلى النوع الذي يلائم المتغير:

```
int height;
cout << "How tall is the room? ";
cin >> height;
```

إظهار محث ما: Displaying a Prompt

- المحث هو رسالة تخبر المستخدم أن يقوم بإدخال البيانات
- دائما يجب أن تستخدم cout لإظهار محث ما قبل كل عبارة cin.

```
cout << "How tall is the room? ";
cin >> height;
```

The `cin` Object

- يمكن استخدام الكائن cin لإدخال أكثر من قيمة واحدة:

```
cin >> height >> width;
```

يجب فصل القيم المتعددة من لوحة المفاتيح بالفراغات.

- الترتيب مهم: تذهب القيمة المدخلة أولا إلى المتغير الأول وهكذا.

The cin Object Gathers Multiple Values in Program 3-2

Program 3-2

```
1  // This program asks the user to enter the length and width of
2  // a rectangle. It calculates the rectangle's area and displays
3  // the value on the screen.
4  #include <iostream>
5  using namespace std;
6
7  int main()
8  {
9     int length, width, area;
10
11    cout << "This program calculates the area of a ";
12    cout << "rectangle.\n";
13    cout << "Enter the length and width of the rectangle ";
14    cout << "separated by a space.\n";
15    cin >> length >> width;
16    area = length * width;
17    cout << "The area of the rectangle is " << area << endl;
18    return 0;
19 }
```

Program Output with Example Input Shown in Bold

```
This program calculates the area of a rectangle.
Enter the length and width of the rectangle separated by a space.
10 20 [Enter]
The area of the rectangle is 200
```

The cin Object Reads Different Data Types in Program 3-3

Program 3-3

```
1  // This program demonstrates how cin can read multiple values
2  // of different data types.
3  #include <iostream>
4  using namespace std;
5
6  int main()
7  {
8     int whole;
9     double fractional;
10    char letter;
11
12    cout << "Enter an integer, a double, and a character: ";
13    cin >> whole >> fractional >> letter;
14    cout << "Whole: " << whole << endl;
15    cout << "Fractional: " << fractional << endl;
16    cout << "Letter: " << letter << endl;
17    return 0;
18 }
```

Program Output with Example Input Shown in Bold

```
Enter an integer, a double, and a character: 4 5.7 b [Enter]
Whole: 4
Fractional: 5.7
Letter: b
```

3.2

التعابير الرياضية

Mathematical Expressions

Mathematical Expressions

- يمكن أن تولد تعابير معقدة باستخدام عدة عوامل رياضية
- يمكن أن يكون التعبير محرفيا Literal أو متغيرا variable أو مجموعة رياضية ذات ثوابت ومتغيرات constants and variables
- يمكن أن تستخدم ضمن الإسناد cout عبارات أخرى:

```
area = 2 * PI * radius;
cout << "border is: " << 2*(l+w);
```

Order of Operations ترتيب العمليات

في التعبير ذو الأكثر من معامل، يتم حساب هذا التعبير بالترتيب التالي:

* / %, in order, left to right

\+ -, in order, left to right

In the expression 2 + 2 * 2 – 2

Order of Operations

Table 3-2 Some Simple Expressions and Their Values

Expression	Value
5 + 2 * 4	13
10 / 2 − 3	2
8 + 12 * 2 − 4	28
4 + 17 % 2 − 1	4
6 − 3 * 2 + 7 − 1	6

Associativity of Operators

- `*, /, %, +, -` associate right to left
- parentheses ( ) can be used to override the order of operations:

```
 2 + 2  *  2 – 2  = 4
(2 + 2) *  2 – 2  = 6
 2 + 2  * (2 – 2) = 2
(2 + 2) * (2 – 2) = 0
```

التجميع بالأقواس
Grouping with Parentheses

Table 3-4 More Simple Expressions and Their Values

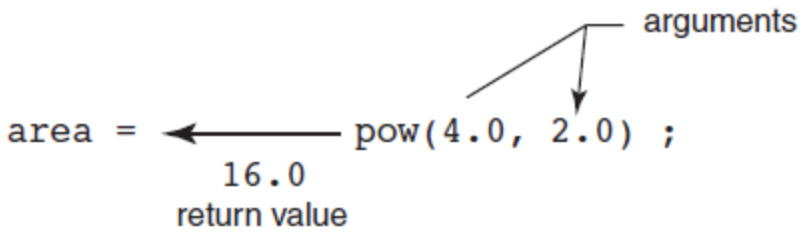
Expression	Value
(5 + 2) * 4	28
10 / (5 − 3)	5
8 + 12 * (6 − 2)	56
(4 + 17) % 2 − 1	0
(6 − 3) * (2 + 7) / 3	9

Algebraic Expressions التعابير الجبرية

- يتطلب الضرب عاملا:

يكتب التعبير *Area=lw* بالشكل

```
Area = l * w;
```



- لا يوجد عامل للأس exponentiation:

$Area=s^2$ is written as `Area = pow(s, 2);`

- يمكن استخدام الأقواس لضمان ترتيب العمليات:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

is written as `m = (y2-y1) /(x2-x1);`

تعابير جبرية
Algebraic Expressions

Table 3-5 Algebraic and C++ Multiplication Expressions

Algebraic Expression	Operation	C++ Equivalent
6*B*	6 times B	6 * B
(3)(12)	3 times 12	3 * 12
4*xy*	4 times x times y	4 * x * y

3.3

متى تخلط التفاح مع البرتقال: تحويل النوع

When You Mix Apples with Oranges: Type Conversion

When You Mix Apples with Oranges: Type Conversion

- تنفذ العمليات بين معاملات operands لها نفس النوع.
- إذا لم تكن من نفس النوع تقوم C++ بتحويل نوع أحدها إلى نوع المعامل الآخر.
- يمكن أن يؤثر ذلك على نتائج الحسابات.

هرمية الأنواع
Hierarchy of Types

Highest:
```
long double
double
float
unsigned long
long
unsigned int
int
```
Lowest:

Ranked by largest number they can hold

قسر (إجبار) النوع Type Coercion

- قسر النوع Type Coercion: هو تحويل آلي لمعامل ما إلى نوع بيانات آخر.
- الترقية Promotion: هي عملية تحويل إلى نوع أعلى.
- إنزال الدرجة (المرتبة) Demotion: هي عملية التحويل إلى نوع أخفض.

قواعد الإجبار Coercion Rules

١) يتم ترقية الأنواع `char, short, unsigned short` بشكل آلي إلى النوع int

٢) عند العمل على قيم ذات أنواع بيانات مختلفة تتم ترقية النوع الأخفض إلى النوع الأعلى.

٣) عند استخدام العامل = يتم تحويل نوع التعبير على الجهة اليمنى إلى نوع المتغير على الجهة اليسرى.

3.4

Overflow and Underflow

Overflow and Underflow

- Occurs when assigning a value that is too large (overflow) or too small (underflow) to be held in a variable
- Variable contains value that is ‘wrapped around’ set of possible values
- Different systems may display a warning/error message, stop the program, or continue execution using the incorrect value

3.5

سبك النوع

Type Casting

Type Casting

- يستخدم لتحويل نوع البيانات يدويا
- يستخدم لتقسيم الفاصلة العائمة باستخدام الأعداد الصحيحة ints:

```
double m;
m = static_cast<double>(y2-y1)
                        /(x2-x1);
```

- Useful to see `int` value of a `char` variable:

```
char ch = 'C';
cout << ch << " is "
     << static_cast<int>(ch);
```

Type Casting in Program 3-9

Program 3-9

```
1  // This program uses a type cast to avoid integer division.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int books;         // Number of books to read
8     int months;        // Number of months spent reading
9     double perMonth;   // Average number of books per month
10
11    cout << "How many books do you plan to read? ";
12    cin >> books;
13    cout << "How many months will it take you to read them? ";
14    cin >> months;
15    perMonth = static_cast<double>(books) / months;
16    cout << "That is " << perMonth << " books per month.\n";
17    return 0;
18 }
```

Program Output with Example Input Shown in Bold

```
How many books do you plan to read? 30 [Enter]
How many months will it take you to read them? 7 [Enter]
That is 4.28571 books per month.
```

C-Style and Prestandard Type Cast Expressions

- 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

3.6

Multiple Assignment and Combined Assignment

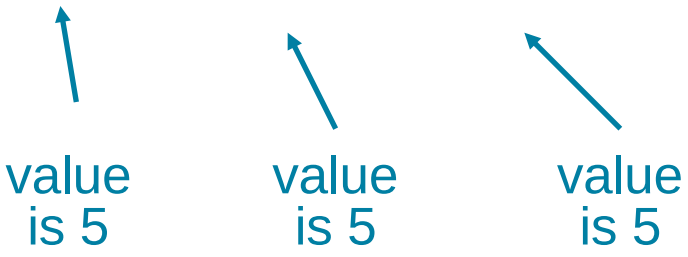
Multiple الاسناد المتعدد والاسناد المجمع Assignment and Combined Assignment

- يمكن استخدام = لإسناد قيمة إلى عدة متغيرات:

```
x = y = z = 5;
```

- Value of = is the value that is assigned
- Associates right to left:

```
x = (y = (z = 5));
```



الإسناد المجمّع: Combined Assignment

- انظر العبارة التالية:

```
sum = sum + 1;
```

إنها تضيف 1 للمتغير **sum**.

عبارات مماثلة أخرى
Other Similar Statements

Table 3-8 (Assume x = 6)

Statement	What It Does	Value of x After the Statement
x = x + 4;	Adds 4 to x	10
x = x − 3;	Subtracts 3 from x	3
x = x * 10;	Multiplies x by 10	60
x = x / 2;	Divides x by 2	3
x = x % 4	Makes x the remainder of x / 4	2

الإسناد المدمج (المجمّع): Combined Assignment

- تقدم عوامل الإسناد المدمجة (المجمّعة) طريقة مختصرة لهذه الأنواع من العبارات.
- العبارة التالية:

```
sum = sum + 1;
```

تكافئ العبارة المدمجة:

```
sum += 1;
```

Combined Assignment Operators

Table 3-9 Combined Assignment Operators

Operator	Example Usage	Equivalent to
+=	x += 5;	x = x + 5;
−=	y −= 2;	y = y − 2;
*=	z *= 10;	z = z * 10;
/=	a /= b;	a = a / b;
%=	c %= 3;	c = c % 3;

3.7

تنسيق (تشكيل) الخرج

Formatting Output

Formatting Output

- يمكنه التحكم بكيفية إظهار الخرج بالنسبة للبيانات العددية أو السلاسل numeric, string data من حيث:
  - size الحجم
  - position الموضع
  - number of digits عدد الخانات
  - يتطلب ملف ترويسة اسمه `iomanip`

معالجات التدفق: Stream Manipulators

- تستخدم للتحكم بكيفية إظهار حقل الخرج output field
- البعض منها يؤثر فقط على القيمة الظاهرة التالية:

– `setw(x)`: print in a field at least x spaces wide. Use more spaces if field is not wide enough

The `setw` Stream Manipulator in Program 3-13

Program 3-13

```
1  // This program displays three rows of numbers.
2  #include <iostream>
3  #include <iomanip>      // Required for setw
4  using namespace std;
5
6  int main()
7  {
8     int num1 = 2897, num2 = 5,    num3 = 837,
9         num4 = 34,   num5 = 7,    num6 = 1623,
10        num7 = 390,  num8 = 3456, num9 = 12;
11
12    // Display the first row of numbers
13    cout << setw(6) << num1 << setw(6)
14         << num2 << setw(6) << num3 << endl;
15
16    // Display the second row of numbers
17    cout << setw(6) << num4 << setw(6)
18         << num5 << setw(6) << num6 << endl;
19
20    // Display the third row of numbers
21    cout << setw(6) << num7 << setw(6)
22         << num8 << setw(6) << num9 << endl;
23    return 0;
24 }
```

Continued…

The `setw` Stream Manipulator in Program 3-13

Program Output

```
 2897     5   837
   34     7  1623
  390  3456    12
```

Stream Manipulators

- البعض منها يؤثر على القيم حتى تتغير مرة أخرى:

  - `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

More Stream Manipulators in Program 3-17

Program 3-17

```
1  // This program asks for sales amounts for 3 days. The total
2  // sales are calculated and displayed in a table.
3  #include <iostream>
4  #include <iomanip>
5  using namespace std;
6
7  int main()
8  {
9       double day1, day2, day3, total;
10
11      // Get the sales for each day.
12      cout << "Enter the sales for day 1: ";
13      cin >> day1;
14      cout << "Enter the sales for day 2: ";
15      cin >> day2;
16      cout << "Enter the sales for day 3: ";
17      cin >> day3;
18
19      // Calculate the total sales.
20      total = day1 + day2 + day3;
21
```

More Stream Manipulators in Program 3-17

```
22      // Display the sales amounts.
23      cout << "\nSales Amounts\n";
24      cout << "-------------\n";
25      cout << setprecision(2) << fixed;
26      cout << "Day 1: " << setw(8) << day1 << endl;
27      cout << "Day 2: " << setw(8) << day2 << endl;
28      cout << "Day 3: " << setw(8) << day3 << endl;
29      cout << "Total: " << setw(8) << total << endl;
30      return 0;
31  }
```

Program Output with Example Input Shown in Bold

```
Enter the sales for day 1: 1321.87 Enter
Enter the sales for day 2: 1869.26 Enter
Enter the sales for day 3: 1403.77 Enter

Sales Amounts
-------------
Day 1:     1321.87
Day 2:     1869.26
Day 3:     1403.77
Total:     4594.90
```

Stream Manipulators

Table 3-12 Stream Manipulators

Stream Manipulator	Description
setw(*n*)	Establishes a print field of *n* spaces.
fixed	Displays floating-point numbers in fixed-point notation.
showpoint	Causes a decimal point and trailing zeros to be displayed, even if there is no fractional part.
setprecision(*n*)	Sets the precision of floating-point numbers.
left	Causes subsequent output to be left-justified.
right	Causes subsequent output to be right-justified.

3.8

التعامل مع المحارف characters ومع كائنات string

Working with Characters and `string` Objects

Working with Characters and `string` Objects

- يمكن أن يسبب استخدام cin مع العامل >> لإدخال السلاسل strings مشاكل مثل:
  - يتجاوز ويتجاهل أي محارف فراغ سابقة leading *whitespace characters* مثل *spaces, tabs, or line breaks*
- لحل هذه المعضلة يمكننا استخدام دالة C++ المسماة **`getline`**.

Using `getline` in Program 3-19

Program 3-19

```
1  // This program demonstrates using the getline function
2  // to read character data into a string object.
3  #include <iostream>
4  #include <string>
5  using namespace std;
6
7  int main()
8  {
9     string name;
10    string city;
11
12    cout << "Please enter your name: ";
13    getline(cin, name);
14    cout << "Enter the city you live in: ";
15    getline(cin, city);
16
17    cout << "Hello, " << name << endl;
18    cout << "You live in " << city << endl;
19    return 0;
20 }
```

Program Output with Example Input Shown in Bold

```
Please enter your name: Kate Smith [Enter]
Enter the city you live in: Raleigh [Enter]
Hello, Kate Smith
You live in Raleigh
```

Working with Characters and `string` Objects

- لقراءة محرف مفرد:
  - 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

Using `cin.get()` in Program 3-21

Program 3-21

```
1  // This program demonstrates three ways
2  // to use cin.get() to pause a program.
3  #include <iostream>
4  using namespace std;
5
6  int main()
7  {
8     char ch;
9
10    cout << "This program has paused. Press Enter to continue.";
11    cin.get(ch);
12    cout << "It has paused a second time. Please press Enter again.";
13    ch = cin.get();
14    cout << "It has paused a third time. Please press Enter again.";
15    cin.get();
16    cout << "Thank you!";
17    return 0;
18 }
```

Program Output with Example Input Shown in Bold

```
This program has paused. Press Enter to continue. [Enter]
It has paused a second time. Please press Enter again. [Enter]
It has paused a third time. Please press Enter again. [Enter]
Thank you!
```

Working with Characters and `string` Objects

- 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'
```

`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;
```

3.9

دوال مكتبة رياضية إضافية

More Mathematical Library Functions

More Mathematical Library Functions

- تتطلب هذه الدوال ملف ترويسة اسمه `cmath`
- تأخذ `double` كدخل وتعيد `double`
- الدوال المستخدمة بشكل شائع:

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

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`

3.10

التتبع اليدوي لبرنامج ما

Hand Tracing a Program

Hand Tracing a Program

- التتبع اليدوي لبرنامج ما: يعمل كما لو أنك الحاسب الذي ينفذ البرنامج بحيث:
  - يتقدم خطوة خطوة وينفذ تعليمة – تعليمة
  - يسجل محتويات المتغيرات بعد تنفيذ العبارة باستخدام مخطط التتبع اليدوي (الجدول)

- Useful to locate logic or mathematical errors

Program 3-27 with Hand Trace Chart

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

```
1  // This program asks for three numbers, then
2  // displays the average of the numbers.
3  #include <iostream>
4  using namespace std;
5  int main()
6  {
7       double num1, num2, num3, avg;
8       cout << "Enter the first number: ";
9       cin >> num1;
10      cout << "Enter the second number: ";
11      cin >> num2;
12      cout << "Enter the third number: ";
13      cin >> num3;
14      avg = num1 + num2 + num3 / 3;
15      cout << "The average is " << avg << endl;
16      return 0;
17 }
```

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.

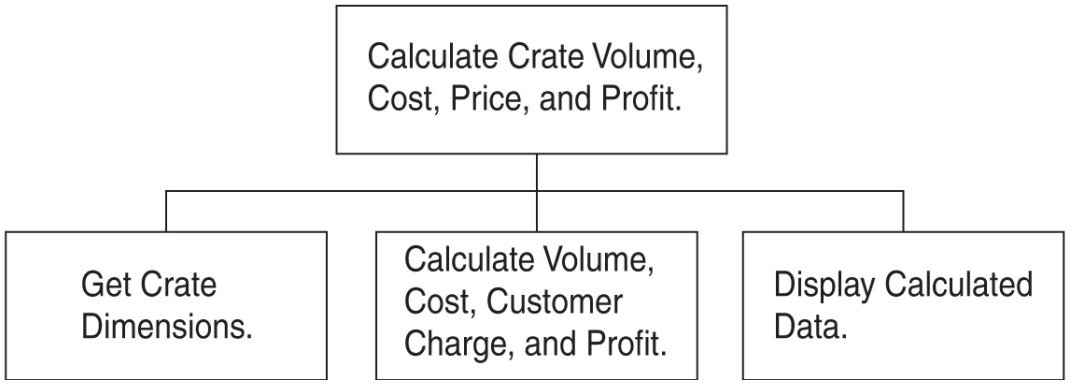
Program Design

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.

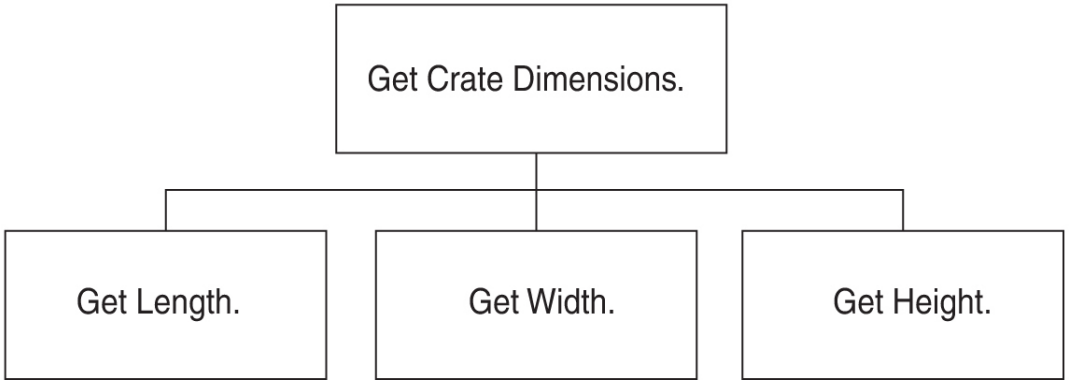
General Hierarchy Chart

Figure 3-7



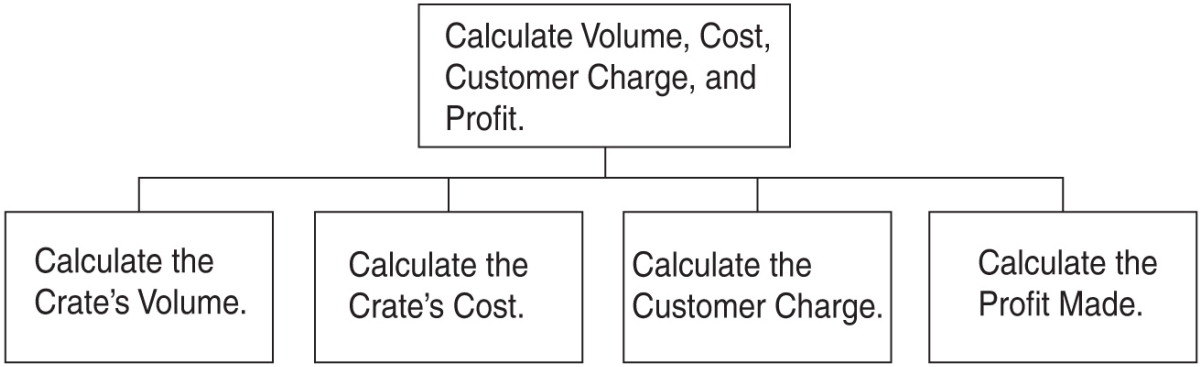
Get Crate Dimensions

Figure 3-8



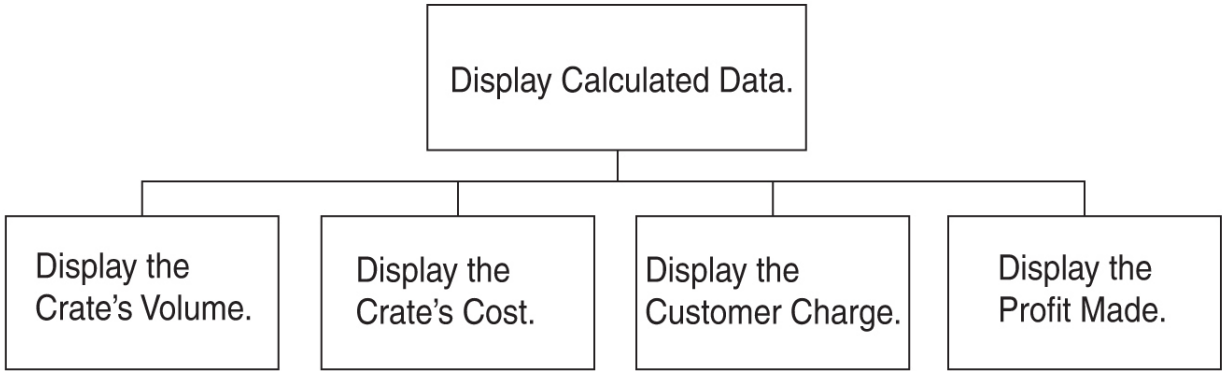
Calculate Volume, Cost, Customer Charge, and Profit

Figure 3-9



Display Calculated Data

Figure 3-10



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:

$$\text{volume} = \text{length} \times \text{width} \times \text{height}$$

$$\text{cost} = \text{volume} \times 0.23$$

$$\text{charge} = \text{volume} \times 0.5$$

$$\text{profit} = \text{charge} - \text{cost}$$

The Program

Program 3-28

```
1  // This program is used by General Crates, Inc. to calculate
2  // the volume, cost, customer charge, and profit of a crate
3  // of any size. It calculates this data from user input, which
4  // consists of the dimensions of the crate.
5  #include <iostream>
6  #include <iomanip>
7  using namespace std;
8
9  int main()
10 {
11    // Constants for cost and amount charged
12    const double COST_PER_CUBIC_FOOT = 0.23;
13    const double CHARGE_PER_CUBIC_FOOT = 0.5;
14
15    // Variables
16    double length,   // The crate's length
17           width,    // The crate's width
18           height,   // The crate's height
19           volume,   // The volume of the crate
20           cost,     // The cost to build the crate
21           charge,   // The customer charge for the crate
22           profit;   // The profit made on the crate
23
24    // Set the desired output formatting for numbers.
25    cout << setprecision(2) << fixed << showpoint;
26
```

Continued…

The Program

```
27      // Prompt the user for the crate's length, width, and height
28      cout << "Enter the dimensions of the crate (in feet):\n";
29      cout << "Length: ";
30      cin >> length;
31      cout << "Width: ";
32      cin >> width;
33      cout << "Height: ";
34      cin >> height;
35
36      // Calculate the crate's volume, the cost to produce it,
37      // the charge to the customer, and the profit.
38      volume = length * width * height;
39      cost = volume * COST_PER_CUBIC_FOOT;
40      charge = volume * CHARGE_PER_CUBIC_FOOT;
41      profit = charge - cost;
42
43      // Display the calculated data.
44      cout << "The volume of the crate is ";
45      cout << volume << " cubic feet.\n";
46      cout << "Cost to build: $" << cost << endl;
47      cout << "Charge to customer: $" << charge << endl;
48      cout << "Profit: $" << profit << endl;
49      return 0;
50  }
```

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
```

Copyright