Introduction to C++ Programming: Solutions

2

What's in a name? that which we call a rose
By any other name would smell as sweet.

—William Shakespeare

When faced with a decision, I always ask, "What would be the most fun?"

—Peggy Walker

"Take some more tea," the March Hare said to Alice, very earnestly. "I've had nothing yet," Alice replied in an offended tone: "so I can't take more." "You mean you can't take less," said the Hatter: "it's very easy to take more than nothing."

—Lewis Carroll

High thoughts must have high language.

—Aristophane

Objectives

In this chapter you'll learn:

- To write simple computer programs in C++.
- To write simple input and output statements.
- To use fundamental types.
- Basic computer memory concepts.
- To use arithmetic operators.
- The precedence of arithmetic operators.
- To write simple decision-making statements.

Student Solution Exercises

2.7 Discuss the meaning of each of the following objects:

a) `std::cin`

ANS: This object refers to the standard input device that is normally connected to the keyboard.

b) `std::cout`

ANS: This object refers to the standard output device that is normally connected to the screen.

2.9 Write a single C++ statement or line that accomplishes each of the following:

a) Print the message `"Enter two numbers"`.

ANS: `cout << "Enter two numbers";`

b) Assign the product of variables `b` and `c` to variable `a`.

ANS: `a = b * c;`

c) State that a program performs a payroll calculation (i.e., use text that helps to document a program).

ANS: `// Payroll calculation program`

d) Input three integer values from the keyboard into integer variables `a`, `b` and `c`.

ANS: `cin >> a >> b >> c;`

2.12 What, if anything, prints when each of the following C++ statements is performed? If nothing prints, then answer "nothing." Assume `x = 2` and `y = 3`.

a) `cout << x;`

ANS: `2`

b) `cout << x + x;`

ANS: `4`

c) `cout << "x=";`

ANS: `x=`

d) `cout << "x = " << x;`

ANS: `x = 2`

e) `cout << x + y << " = " << y + x;`

ANS: `5 = 5`

f) `z = x + y;`

ANS: nothing.

g) `cin >> x >> y;`

ANS: nothing.

h) `// cout << "x + y = " << x + y;`

ANS: nothing (because it is a comment).

i) `cout << "\n";`

ANS: A newline is output which positions the cursor at the beginning of the next line on the screen.

2.14 Given the algebraic equation $y = ax^3 + 7$, which of the following, if any, are correct C++ statements for this equation?

a) `y = a * x * x * x + 7;`

b) `y = a * x * x * ( x + 7 );`

c) `y = ( a * x ) * x * ( x + 7 );`

d) `y = (a * x) * x * x + 7;`

e) `y = a * ( x * x * x ) + 7;`

f) `y = a * x * ( x * x + 7 );`

ANS: Parts (a), (d) and (e).

2.17 *(Printing)* Write a program that prints the numbers 1 to 4 on the same line with each pair of adjacent numbers separated by one space. Do this several ways:

a) Using one statement with one stream insertion operator.
b) Using one statement with four stream insertion operators.
c) Using four statements.

ANS:

```
// Exercise 2.17 Solution: ex02_17.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   // Part a
   cout << "1 2 3 4\n";

   // Part b
   cout << "1 " << "2 " << "3 " << "4\n";

   // Part c
   cout << "1 ";
   cout << "2 ";
   cout << "3 ";
   cout << "4" << endl;
} // end main
```

```
1 2 3 4
1 2 3 4
1 2 3 4
```

2.18 *(Comparing Integers)* Write a program that asks the user to enter two integers, obtains the numbers from the user, then prints the larger number followed by the words "is larger." If the numbers are equal, print the message "These numbers are equal."

ANS:

```
// Exercise 2.18 Solution: ex02_18.cpp
#include <iostream> // allows program to perform input and output
using namespace std;

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user

   cout << "Enter two integers: "; // prompt user for data
   cin >> number1 >> number2; // read two integers from user

   if ( number1 == number2 )
      cout << "These numbers are equal." << endl;

   if ( number1 > number2 )
```

```
      cout << number1 << " is larger." << endl;

   if ( number2 > number1 )
      cout << number2 << " is larger." << endl;
} // end main
```

```
Enter two integers: 22 8
22 is larger.
```

```
Enter two integers: 8 22
22 is larger.
```

```
Enter two integers: 22 22
These numbers are equal.
```

2.23 *(Largest and Smallest Integers)* Write a program that reads in five integers and determines and prints the largest and the smallest integers in the group. Use only the programming techniques you learned in this chapter.

ANS:

```
// Exercise 2.23 Solution: ex02_23.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user
   int number3; // third integer read from user
   int number4; // fourth integer read from user
   int number5; // fifth integer read from user
   int smallest; // smallest integer read from user
   int largest; // largest integer read from user

   cout << "Enter five integers: "; // prompt user for data
   cin >> number1 >> number2 >> number3 >> number4 >> number5;

   largest = number1; // assume first integer is largest
   smallest = number1; // assume first integer is smallest

   if ( number2 > largest ) // is number2 larger?
      largest = number2; // number2 is new largest

   if ( number3 > largest ) // is number3 larger?
      largest = number3; // number3 is new largest

   if ( number4 > largest ) // is number4 larger?
      largest = number4; // number4 is new largest

   if ( number5 > largest ) // is number5 larger?
      largest = number5; // number5 is new largest

   if ( number2 < smallest ) // is number2 smaller?
      smallest = number2; // number2 is new smallest

   if ( number3 < smallest ) // is number3 smaller?
      smallest = number3; // number3 is new smallest

   if ( number4 < smallest ) // is number4 smaller?
      smallest = number4; // number4 is new smallest

   if ( number5 < smallest ) // is number5 smaller?
      smallest = number5; // number5 is new smallest

   cout << "Largest is " << largest
      << "\nSmallest is " << smallest << endl;
} // end main
```

```
Enter five integers: 88 22 8 78 21
Largest is 88
Smallest is 8
```

2.27 *(Integer Equivalent of a Character)* Here is a peek ahead. In this chapter you learned about integers and the type `int`. C++ can also represent uppercase letters, lowercase letters and a considerable variety of special symbols. C++ uses small integers internally to represent each different character. The set of characters a computer uses and the corresponding integer representations for those characters are called that computer's **character set**. You can print a character by enclosing that character in single quotes, as with

```
cout << 'A'; // print an uppercase A
```

You can print the integer equivalent of a character using `static_cast` as follows:

```
cout << static_cast< int >( 'A' ); // print 'A' as an integer
```

This is called a **cast** operation (we formally introduce casts in Chapter 4). When the preceding statement executes, it prints the value 65 (on systems that use the **ASCII character set**). Write a program that prints the integer equivalent of a character typed at the keyboard. Store the input in a variable of type `char`. Test your program several times using uppercase letters, lowercase letters, digits and special characters (like `$`).

ANS:

```
// Exercise 2.27 Solution: ex02_27.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   char symbol; // char read from user

   cout << "Enter a character: "; // prompt user for data
   cin >> symbol; // read the character from the keyboard

   cout << symbol << "'s integer equivalent is "
      << static_cast< int >( symbol ) << endl;
} // end main
```

```
Enter a character: A
A's integer equivalent is 65
```

```
Enter a character: B
B's integer equivalent is 66
```

```
Enter a character: a
a's integer equivalent is 97
```

```
Enter a character: 7
7's integer equivalent is 55
```

```
Enter a character: $
$'s integer equivalent is 36
```

2.29 *(Table)* Using only the techniques you learned in this chapter, write a program that calculates the squares and cubes of the integers from 0 to 10 and uses tabs to print the following neatly formatted table of values:

```
integer square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```

ANS:

```cpp
// Exercise 2.29 Solution: ex02_29.cpp
#include <iostream> // allows program to perform input and output
using namespace std;

int main()
{
   int number; // integer to square and cube

   number = 0; // set number to 0
   cout << "integer\tsquare\tcube\n"; // output column heads

   // output the integer, its square and its cube
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 1; // set number to 1
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 2; // set number to 2
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 3; // set number to 3
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 4; // set number to 4
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 5; // set number to 5
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 6; // set number to 6
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 7; // set number to 7
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 8; // set number to 8
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 9; // set number to 9
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 10; // set number to 10
   cout << number << '\t' << number * number << '\t'
      << number * number * number << endl;
} // end main
```

```
integer square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```