4.12 Compound Assignment Operators

The **compound assignment operators** enable you to abbreviate assignment expressions. For example, you can abbreviate the statement

```
c = c + 3; // adds 3 to c
```

with the **addition compound assignment operator**, `+=`, as

```
c += 3; // adds 3 to c more concisely
```

The `+=` operator adds the value of the expression on its right to the value of the variable on its left and stores the result in the variable on the left. Thus, the assignment expression `c+=3` adds `3` to `c`. In general, statements like

```
variable = variable operator expression;
```

where *operator* is one of the binary operators `+`, `-`, `*`, `/` or `%` (or others we discuss later in the text) and the same variable name is used can be written in the form

```
variable operator=  expression;
```

Figure 4.13 shows the arithmetic compound assignment operators, sample expressions using the operators and explanations of what the operators do. Be sure to do Self-Review Exercise 4.1(h) to learn about a subtle feature of compound assignment operators.

Assignment operator	Sample expression	Explanation	Assigns
Assume: `int c = 3, d = 5, e = 4, f = 6, g = 12;`			
`+=`	`c += 7`	`c = c + 7`	`10` to `c`
`-=`	`d -= 4`	`d = d - 4`	`1` to `d`
`*=`	`e *= 5`	`e = e * 5`	`20` to `e`
`/=`	`f /= 3`	`f = f / 3`	`2` to `f`
`%=`	`g %= 9`	`g = g % 9`	`3` to `g`

Fig. 4.13

Arithmetic compound assignment operators.