Supplement: The & and | Operators
For Introduction to Programming with C++
By Y. Daniel Liang

C++ also provides the **&** and **|** operators. The **&** operator works exactly the same as the **&&** operator, and the **|** operator works exactly the same as the **||** operator with one exception: the **&** and **|** operators always evaluate both operands. Therefore, **&** is referred to as the *unconditional AND* operator, and **|** is referred to as the *unconditional OR* operator. In some rare situations when needed, you can use the **&** and **|** operators to guarantee that the right-hand operand is evaluated regardless of whether the left-hand operand is **true** or **false**. For example, the expression **(width < 2) & (height-- < 2)** guarantees that **(height-- < 2)** is evaluated. Thus the variable **height** will be decremented regardless of whether **width** is less than 2 or not.

> TIP
> Avoid using the **&** and **|** operators. The benefits of the **&** and **|** operators are marginal. Using them will make the program difficult to read and could cause errors. For example, the expression **(x != 0) & (100 / x > 1)** results in a runtime error if **x** is 0. However, **(x != 0) && (100 / x > 1)** is fine. If **x** is 0, **(x != 0)** is false. Since **&&** is a short-circuit operator, C++ does not evaluate **(100 / x > 1)** and returns the result as false for the entire expression **(x != 0) && (100 / x > 1)**.

> **NOTE**
> The **&** and **|** operators can also apply to bitwise operations. See Supplement III.K, "Bit Operations," for details.