Programming Language (630203)
Fall 2010/2011 – Lecture Notes # 5

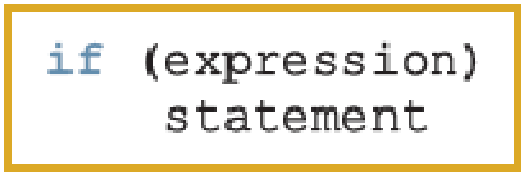
Selection Control Structures in C++

Objectives of the Lecture

- **One-Way Selection.**
- **Two-Way Selection.**
- **Compound (Block of) Statements.**

One-Way Selection

The syntax of one-way selection is:



```
if (expression)
    statement
```

- The statement is executed if the value of the expression is true
- The statement is bypassed if the value is false; program goes to the next statement
- **if** is a reserved word

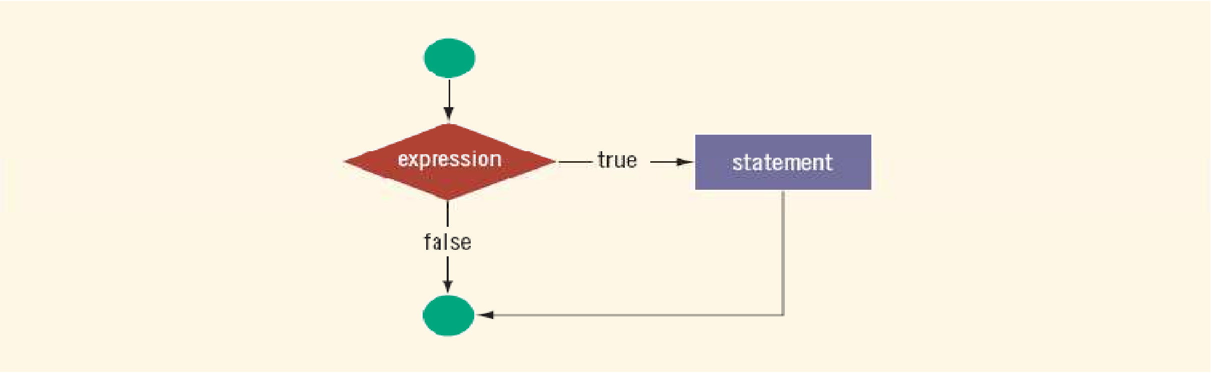
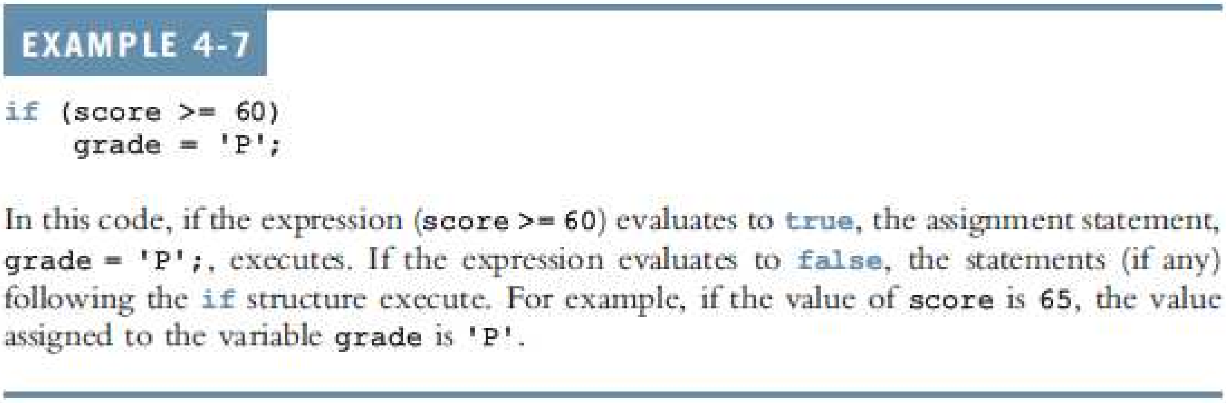


FIGURE 4-2 One-way selection



EXAMPLE 4-7

```
if (score >= 60)
    grade = 'P';
```

In this code, if the expression (score >= 60) evaluates to true, the assignment statement, grade = 'P';, executes. If the expression evaluates to false, the statements (if any) following the if structure execute. For example, if the value of score is 65, the value assigned to the variable grade is 'P'.

EXAMPLE 4-8

The following C++ program finds the absolute value of an integer.

```
//Program: Absolute value of an integer

#include <iostream>

using namespace std;

int main()
{
    int number, temp;

    cout << "Line 1: Enter an integer: ";          //Line 1
    cin >> number;                                 //Line 2
    cout << endl;                                  //Line 3

    temp = number;                                 //Line 4

    if (number < 0)                                //Line 5
        number = -number;                          //Line 6

    cout << "Line 7: The absolute value of "
         << temp << " is " << number << endl;     //Line 7

    return 0;
}
```

Sample Run: In this sample run, the user input is shaded.

```
Line 1: Enter an integer: -6734
Line 7: The absolute value of -6734 is 6734
```

EXAMPLE 4-9

Consider the following statement:

```
if score >= 60      //syntax error
   grade = 'P';
```

This statement illustrates an incorrect version of an `if` statement. The parentheses around the logical expression are missing, which is a syntax error.

EXAMPLE 4-10

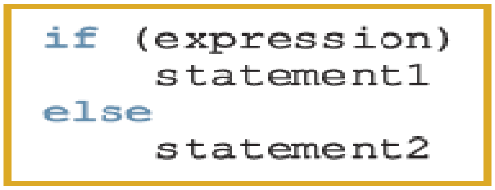
Consider the following C++ statements:

```
if (score >= 60);           //Line 1
    grade = 'P';            //Line 2
```

Because there is a semicolon at the end of the expression (see Line 1), the `if` statement in Line 1 terminates. The action of this `if` statement is null, and the statement in Line 2 is not part of the `if` statement in Line 1. Hence, the statement in Line 2 executes regardless of how the `if` statement evaluates.

Two-Way Selection

Two-way selection takes the form:



```
if (expression)
     statement1
else
     statement2
```

- ➢ If expression is true, statement1 is executed; otherwise, statement2 is executed
  - o statement1 and statement2 are any C++ statements
- ➢ **else** is a reserved word.

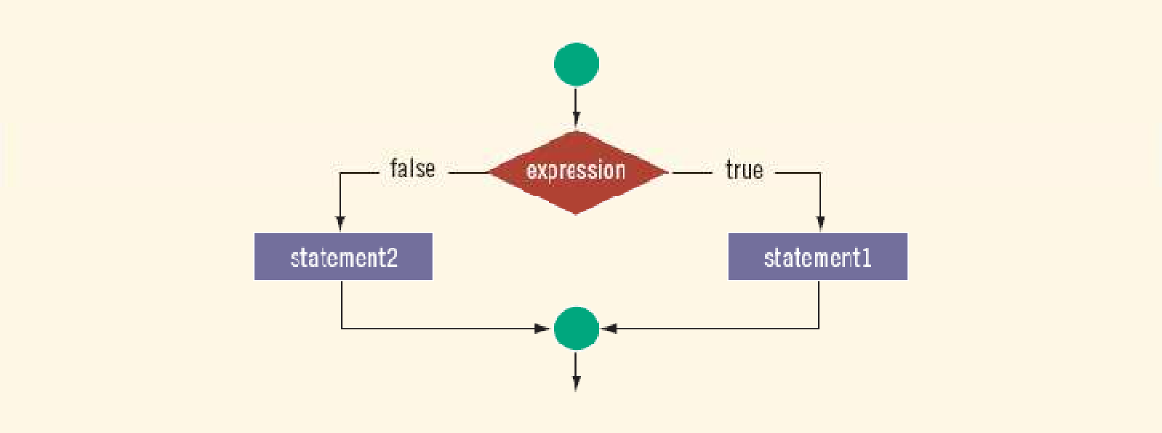
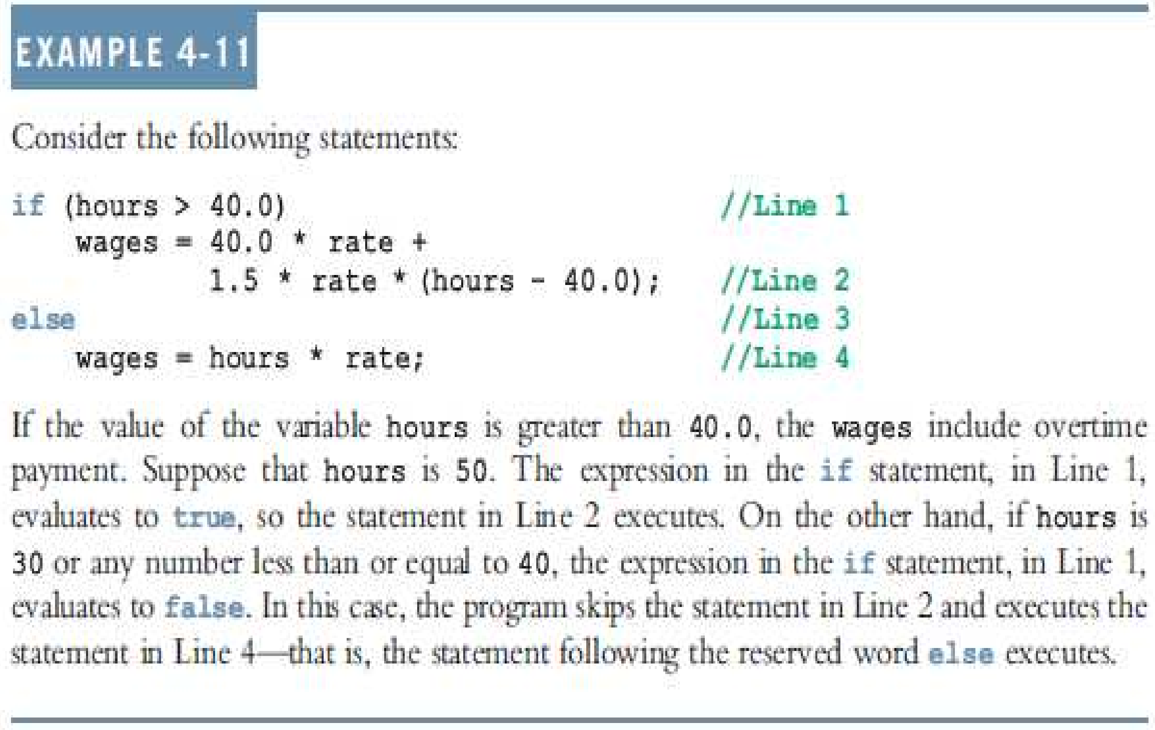


FIGURE 4-3 Two-way selection

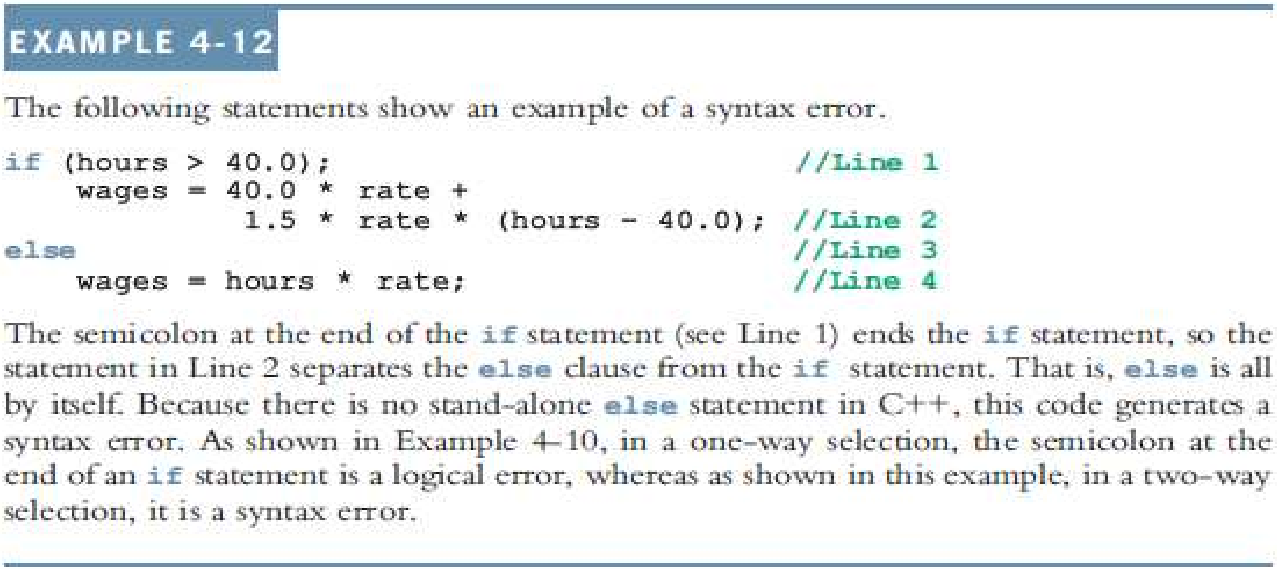


EXAMPLE 4-11

Consider the following statements:

```
if (hours > 40.0)                           //Line 1
    wages = 40.0 * rate +
            1.5 * rate * (hours - 40.0);    //Line 2
else                                        //Line 3
    wages = hours * rate;                   //Line 4
```

If the value of the variable hours is greater than 40.0, the wages include overtime payment. Suppose that hours is 50. The expression in the if statement, in Line 1, evaluates to true, so the statement in Line 2 executes. On the other hand, if hours is 30 or any number less than or equal to 40, the expression in the if statement, in Line 1, evaluates to false. In this case, the program skips the statement in Line 2 and executes the statement in Line 4—that is, the statement following the reserved word else executes.



EXAMPLE 4-12

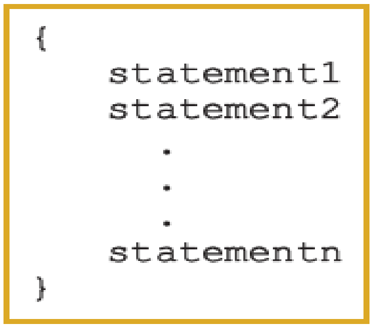
The following statements show an example of a syntax error.

```
if (hours > 40.0);                            //Line 1
    wages = 40.0 * rate +
              1.5 * rate * (hours - 40.0);    //Line 2
else                                          //Line 3
    wages = hours * rate;                     //Line 4
```

The semicolon at the end of the if statement (see Line 1) ends the if statement, so the statement in Line 2 separates the else clause from the if statement. That is, else is all by itself. Because there is no stand-alone else statement in C++, this code generates a syntax error. As shown in Example 4-10, in a one-way selection, the semicolon at the end of an if statement is a logical error, whereas as shown in this example, in a two-way selection, it is a syntax error.

Compound (Block of) Statements

Compound statement (block of statements):



```
{
    statement1
    statement2
       .
       .
       .
    statementn
}
```

A compound statement is a single statement

```
if (age > 18)
{
 cout << "Eligible to vote." << endl;
 cout << "No longer a minor." << endl;
}
else
{
 cout << "Not eligible to vote." << endl;
 cout << "Still a minor." << endl;
}
```