Programming Language (630203)
Fall 2010/2011 – Lecture Notes # 6

Selection Control Structures in C++

Objectives of the Lecture

- **Multiple Selections: Nested if structure**
- **Comparing if…else Statements with a Series of if Statements**
- **Confusion Between the Equality (==) and Assignment (=) Operators**
- **Conditional Operator (?:)**

Multiple Selections: Nested if structure

- **Nesting**: one control statement in another.
- You can have if statements inside other if statements inside other if or else statements, and so on.
- Every else always belongs to the closest if.

Example 1: Create a program that states whether an input number is positive, negative, or zero.
Clearly, we have more than two options, so a single if statement will not work. Instead, I use a nested if construction as follows:

```
#include <iostream>
using namespace std;
int main()
{
   double x;   // the input number
   cout << "Enter the number: ";
   cin >> x;
   if (x == 0)
      cout << "is zero";
   else
      if (x > 0)
         cout << "is positive";
      else
         cout << "is negative";
   return 0;
}
```

Example 2: Suppose we want to create a program that takes as input a grade in percent, and produces as output a letter grade.
The program clearly has multiple choices to make:

```
#include <iostream>
using namespace std;
```

```
int main()
{
   double x;
   cout << "Enter the grade in percent: ";
   cin >> x;
if (x >= 90)
      cout << "A";
   else if (x >=80)
      cout << "B";
   else if (x >=70)
      cout << "C";
   else if (x >=60)
      cout << "D";
   else
      cout << "F";
   return 0;
}
```

EXAMPLE 4-15

Suppose that `balance` and `interestRate` are variables of type `double`. The following statements determine the `interestRate` depending on the value of the `balance`.

```
if (balance > 50000.00)                  //Line 1
    interestRate = 0.07;                 //Line 2
else                                     //Line 3
    if (balance >= 25000.00)             //Line 4
        interestRate = 0.05;             //Line 5
    else                                 //Line 6
        if (balance >= 1000.00)          //Line 7
            interestRate = 0.03;         //Line 8
        else                             //Line 9
            interestRate = 0.00;         //Line 10
```

To avoid excessive indentation, the code in Example 4-15 can be rewritten as follows:

```
if (balance > 50000.00)                  //Line 1
    interestRate = 0.07;                 //Line 2
else if (balance >= 25000.00)            //Line 3
    interestRate = 0.05;                 //Line 4
else if (balance >= 1000.00)             //Line 5
    interestRate = 0.03;                 //Line 6
else                                     //Line 7
    interestRate = 0.00;                 //Line 8
```

EXAMPLE 4-16

Assume that `score` is a variable of type `int`. Based on the value of `score`, the following code outputs the grade.

```
if (score >= 90)
    cout << "The grade is A." << endl;
else if (score >= 80)
    cout << "The grade is B." << endl;
else if (score >= 70)
    cout << "The grade is C." << endl;
else if (score >= 60)
    cout << "The grade is D." << endl;
else
    cout << "The grade is F." << endl;
```

Comparing if…else Statements with a Series of if Statements

a.

```
if (month == 1)                          //Line 1
    cout << "January" << endl;           //Line 2
else if (month == 2)                     //Line 3
    cout << "February" << endl;          //Line 4
else if (month == 3)                     //Line 5
    cout << "March" << endl;             //Line 6
else if (month == 4)                     //Line 7
    cout << "April" << endl;             //Line 8
else if (month == 5)                     //Line 9
    cout << "May" << endl;               //Line 10
else if (month == 6)                     //Line 11
    cout << "June" << endl;              //Line 12
```

b.

```
if (month == 1)
    cout << "January" << endl;
if (month == 2)
    cout << "February" << endl;
if (month == 3)
    cout << "March" << endl;

if (month == 4)
    cout << "April" << endl;
if (month == 5)
    cout << "May" << endl;
if (month == 6)
    cout << "June" << endl;
```

Confusion Between the Equality (==) and Assignment (=) Operators

- C++ allows you to use any expression that can be evaluated to either true or false as an expression in the if statement:

```
if (x = 5)
    cout << "The value is five." << endl;
```

- The appearance of = in place of ==
  - It is not a syntax error
  - It is a logical error

Conditional Operator (?:)

- Conditional operator **(?:)** takes three arguments
  - Ternary operator
- Syntax for using the conditional operator:

```
expression1 ? expression2 : expression3
```

- If expression1 is true, the result of the conditional expression is expression2 Otherwise, the result is expression3