Programming In C++

Lecture 5 if and switch case statements

Prepared By Dr. Ali Al-Sabaawi

Nineveh University- Faculty of IT- department of software

If statement

- If the value of the expression inside the parentheses is true, the very next statement is executed. Otherwise, it is skipped.

```
if (expression)
    statement;
```

```
int main()
{
        int x;
        cin>>x;
        if(x % 2 ==0)
            cout<<“x is even”;
        else
            cout<<“x is odd”;
}
```

else is optional, it can be added in certain cases.

Expanding if statement

- The if statement can conditionally execute a block of statements enclosed in braces.

```
if (expression)
{
    statement;
    statement;
    // Place as many statements here as necessary.
}
```

Expanding if statement

```
3   int main()
4   {
5      const int HIGH_SCORE = 95;       // A high score is 95 or greater
6      int score1, score2, score3;      // To hold three test scores
7      double average;                  // TO hold the average score
8      cout << "Enter 3 test scores and I will average them: ";
9      cin >> score1 >> score2 >> score3;
10      average = (score1 + score2 + score3) / 3.0;
11     cout << "Your average is " << average << endl;
12      if (average > HIGH_SCORE)
13     {
14        cout << "Congratulations!\n";
15        cout << "That's a high score.\n";
16        cout << "You deserve a pat on the back!\n";
17     }
18     return 0;
19  }
```

Don’t forget the braces

```
3   int main()
4   {
5      const int HIGH_SCORE = 95;       // A high score is 95 or greater
6      int score1, score2, score3;      // To hold three test scores
7      double average;                  // TO hold the average score
8      cout << "Enter 3 test scores and I will average them: ";
9      cin >> score1 >> score2 >> score3;
10      average = (score1 + score2 + score3) / 3.0;
11     cout << "Your average is " << average << endl;
12      if (average > HIGH_SCORE)
13
14        cout << "Congratulations!\n";
15        cout << "That's a high score.\n";
16        cout << "You deserve a pat on the back!\n";
17
18     return 0;
19  }
```

If statement

Exercise 1:
Write an if statement that performs the following logic: if the variable sales is greater than 50,000, then assign 0.25 to the commission Rate variable, and assign 250 to the bonus variable.

Exercise 2:
Write an if/else statement that assigns 1 to x if y is equal to 100. Otherwise it should assign 0 to x.

Nested If

The if/else if statement tests a series of conditions.

```
if (expression_1)
{
   statement
   statement
   etc.
}
else if (expression_2)
{
   statement
   statement
   etc.
}
Insert as many else if clauses as necessary
else
{
   statement
   statement
   etc.
}
```

If `expression_1` *is true these statements are executed, and the rest of the structure is ignored.*

Otherwise, if `expression_2` *is true these statements are executed, and the rest of the structure is ignored.*

These statements are executed if none of the expressions above are true.

Nested If

Write a program to display the grade of a student according to his/her score.

Test Score	Grade
90 and above	A
80–89	B
70–79	C
60–69	D
Below 60	F

```
3   int main()
4   {
5      int testScore; // To hold a numeric test score
6      cout << "Enter your numeric test score and I will\n"
7      cin >> testScore;
8      // Determine the letter grade.
9      if (testScore >= 90)
10        cout << "Your grade is A.\n";
11     else if (testScore >= 80)
12        cout << "Your grade is B.\n";
13     else if (testScore >= 70)
14        cout << "Your grade is C.\n";
15     else if (testScore >= 60)
16        cout << "Your grade is D.\n";
17     else
18       cout << "Your grade is F.\n";
19     return 0;
20  }
```

Checking numeric ranges

Logical operators are effective for determining whether a number is in or out of a range.

For example, the following if statement checks the value in x to determine whether it is in the range of 20 through 40:

```
if (x >= 20 && x <= 40)
   cout << x << " is in the acceptable range.\n";
```

NOTE: C++ does not allow you to check numeric ranges with expressions such as `5 < x < 20`. Instead, you must use a logical operator to connect two relational expressions, as previously discussed.

Comparing characters and strings

Example, a program to differentiate an input whether it's upper case, lower case, digit, or something else

```
3   int main()
4   {
5       char ch;
6        // Get a character from the user.
7       cout << "Enter a digit or a letter: ";
8       cin>>ch;
9       // Determine what the user entered.
10      if (ch >= '0' && ch <= '9')
11          cout << "You entered a digit.\n";
12      else if (ch >= 'A' && ch <= 'Z')
13          cout << "You entered an uppercase letter.\n";
14      else if (ch >= 'a' && ch <= 'z')
15          cout << "You entered a lowercase letter.\n";
16      else
17          cout << "That is not a digit or a letter.\n";
18
19      return 0;
20  }
```

Comparing characters and strings

Write a program to convert the uppercase letter to lowercase and vise versa.

Conditional Operator

The conditional operator is powerful and unique. It provides a shorthand method of expressing a simple if/else statement. The operator consists of the question-mark (?) and the colon (:). Its format is:

```
expression ? expression : expression;
```

```
x < 0 ? y = 10 : z = 20;
```

It is similar to the following code

```
if (x < 0)
    y = 10;
else
    z = 20;
```

Switch statement

The switch statement lets the value of a variable or expression determine where the program will branch. The format of the switch statement:

```
switch (IntegerExpression)
{
   case ConstantExpression:
        // place one or more
        // statements here

   case ConstantExpression:
        // place one or more
        // statements here

    // case statements may be repeated as many
    // times as necessary

   default:
        // place one or more
        // statements here
}
```

WARNING! The expression of each `case` statement in the block must be unique.

NOTE: The expression following the word `case` must be an integer literal or constant. It cannot be a variable, and it cannot be an expression such as `x < 22` or `n == 50`.

Switch statement

```
1  // The switch statement in this program tells the user something
2  // he or she already knows: the data just entered!
3  #include <iostream>
4  using namespace std;
5
6  int main()
7  {
8       char choice;
9
10      cout << "Enter A, B, or C: ";
11      cin >> choice;
12      switch (choice)
13      {
14           case 'A': cout << "You entered A.\n";
15                     break;
16           case 'B': cout << "You entered B.\n";
17                     break;
18           case 'C': cout << "You entered C.\n";
19                     break;
20           default:  cout << "You did not enter A, B, or C!\n";
21      }
22      return 0;
23 }
```

(program continues)

Switch statement

Class work:

1- Write a program accepts a digit (0-9) and print the corresponding name of the given value.

2- Write a program to read two numbers and an arithmetic operation. The program implements the operation between the numbers.

HW:

Write a program to read a sequence of a month and print the corresponding name of that month.

The End