

1. Demonstrates the Conditional Operator

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **OperatorConditional**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create a **C++ File (.cpp)** and name it as **main.cpp** by selecting **Project → Add New Item**, and then type the following code inside.

```
// Demonstrates the conditional operator
#include <iostream>
using namespace std;

void main()
{
    int x, y, z;

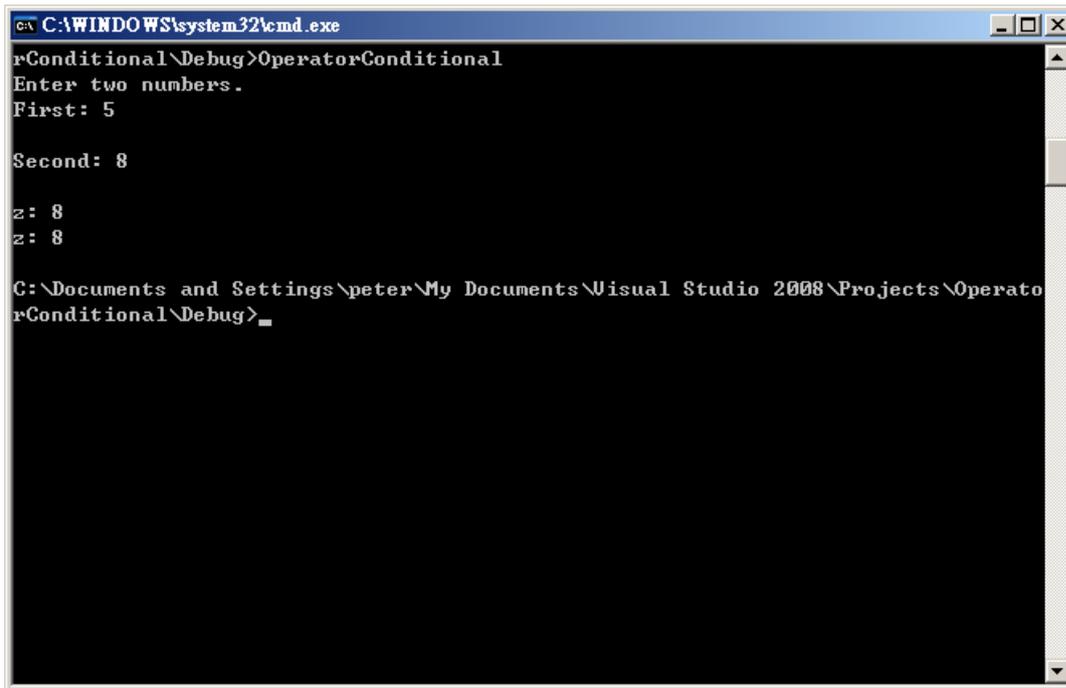
    cout << "Enter two numbers.\n";
    cout << "First: ";
    cin >> x;
    cout << "\nSecond: ";
    cin >> y;
    cout << "\n";

    if (x > y)
        z = x;
    else
        z = y;

    cout << "z: " << z;
    cout << "\n";

    z = (x > y) ? x : y;
    cout << "z: " << z;
    cout << "\n";
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it.
- Three integer variables are created: x, y, and z. The first two are given values by the user. The if statement tests to see which is larger and assigns the larger value to z. This value is printed.
 - The conditional operator makes the same test and assigns z the larger value. It is read like this: "If x is greater than y, return the value of x; otherwise, return the value of y." The value returned is assigned to z. That value is printed. As you can see, the conditional statement is a shorter equivalent to the if...else statement.



```
c:\WINDOWS\system32\cmd.exe
rConditional\Debug>OperatorConditional
Enter two numbers.
First: 5

Second: 8

z: 8
z: 8

C:\Documents and Settings\peter\My Documents\Visual Studio 2008\Projects\OperatorConditional\Debug>_
```

2. Demonstrates the Operator: IF

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **OperatorIf**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create a **C++ File (.cpp)** and name it as **main.cpp** by selecting **Project** → **Add New Item**, and then type the following code inside.

```
// A demonstration of branching based on relational operators.
// demonstrates if statement used with relational operators
#include <iostream>
using namespace std;

void main()
{
    int mu_score, arsenal_score;

    cout << "Enter the score for the Manchester United: ";
    cin >> mu_score;
    cout << "\nEnter the score for the Arsenal: ";
    cin >> arsenal_score;
    cout << "\n";

    if (mu_score > arsenal_score)
        cout << "Go MU!\n";

    if (mu_score < arsenal_score)
    {
        cout << "Go Arsenal!\n";
    }

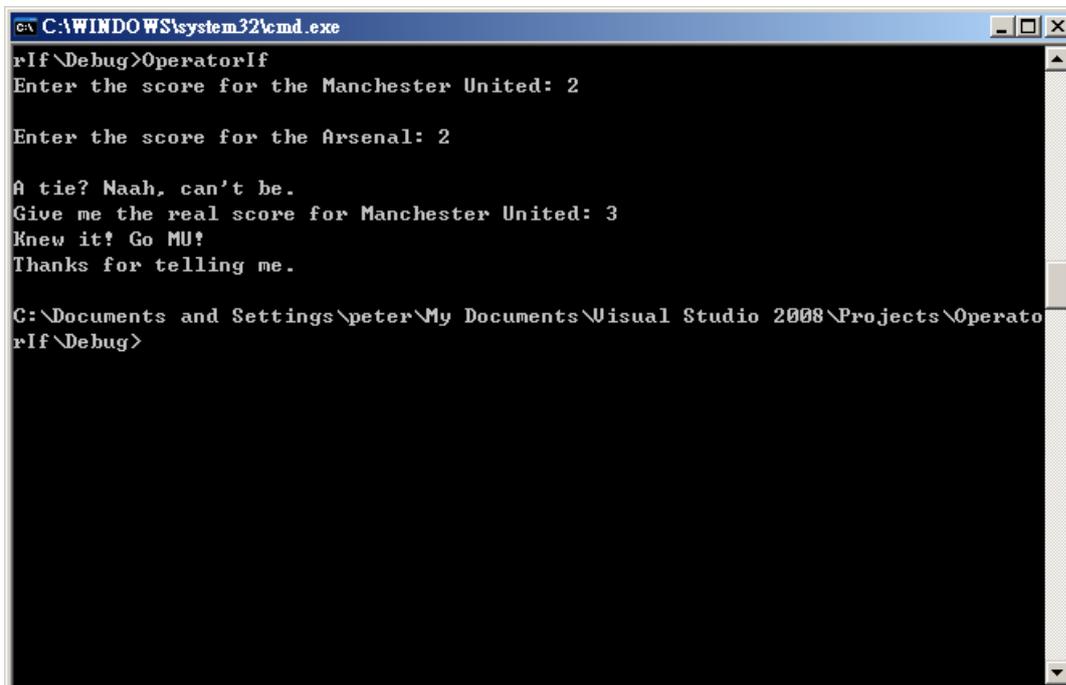
    if (mu_score == arsenal_score)
    {
        cout << "A tie? Naah, can't be.\n";
        cout << "Give me the real score for Manchester United: ";
        cin >> mu_score;
    }
}
```

```
    if (mu_score > arsenal_score)
        cout << "Knew it! Go MU!";

    if (arsenal_score > mu_score)
        cout << "Knew it! Go Arsenal!";

    if (mu_score == arsenal_score)
        cout << "Wow, it really was a tie!";
}
cout << "\nThanks for telling me.\n";
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it.
- This program asks for user input of scores for two soccer teams, which are stored in integer variables. The variables are compared in if statement.
 - If one score is higher than the other, an informational message is printed. If the scores are equal, the block of code is entered. The second score is requested again, and then the scores are compared again.
 - In this example, getting a true result in one if statement does not stop other if statements from being tested..



```
C:\WINDOWS\system32\cmd.exe
rIf\Debug>OperatorIf
Enter the score for the Manchester United: 2

Enter the score for the Arsenal: 2

A tie? Naah, can't be.
Give me the real score for Manchester United: 3
Knew it! Go MU!
Thanks for telling me.

C:\Documents and Settings\peter\My Documents\Visual Studio 2008\Projects\OperatorIf\Debug>
```

3. Demonstrates the Operator: Nested IF

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **OperatorNestedIf**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create a **C++ File (.cpp)** and name it as **main.cpp** by selecting **Project** → **Add New Item**, and then type the following code inside.

```
// A complex nested if statement
// Ask for two numbers
// Assign the numbers to big_number and small_number
// If big_number is bigger than small_number, see if they are evenly divisible
// If they are, see if they are the same number
#include <iostream>
using namespace std;

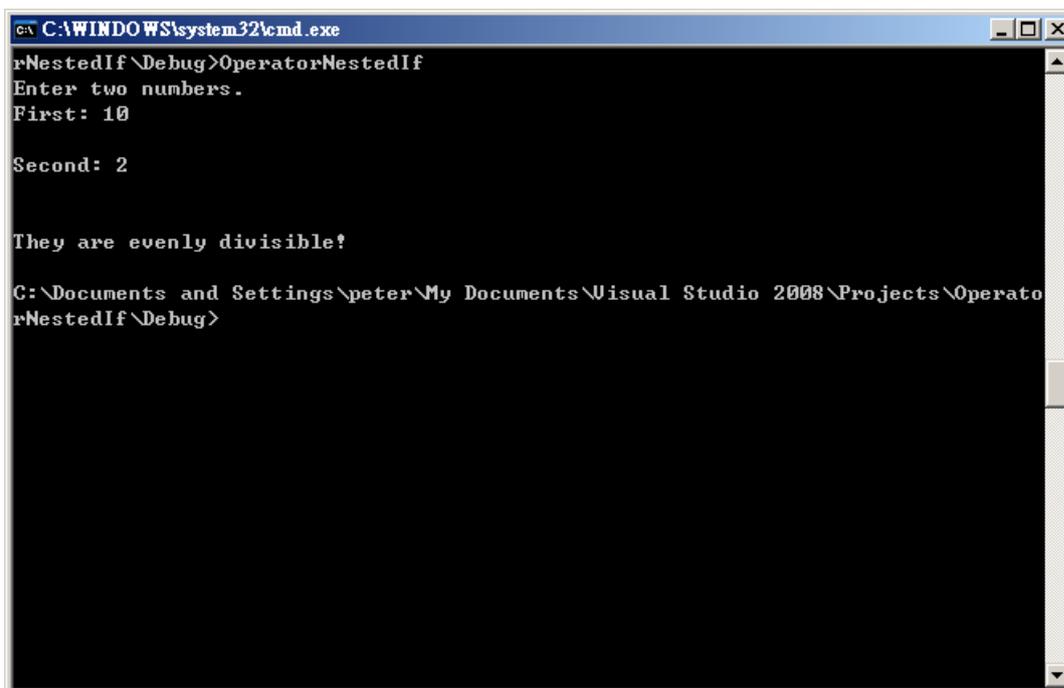
void main()
{
    int first_number, second_number;

    cout << "Enter two numbers.\nFirst: ";
    cin >> first_number;
    cout << "\nSecond: ";
    cin >> second_number;
    cout << "\n\n";

    if (first_number >= second_number)
    {
        if ( (first_number % second_number) == 0)    // evenly divisible?
        {
            if (first_number == second_number)
                cout << "They are the same!\n";
            else
                cout << "They are evenly divisible!\n";
        }
        else
            cout << "They are not evenly divisible!\n";
    }
    else
        cout << "Hey! The second one is larger!\n";
}
```

3.

4. Use **Build Solution** to compile and build the solution, and then execute it.
 - Two numbers are prompted for one at a time, and then compared. The first if statement checks to ensure that the first number is greater than or equal to the second. If not, the else clause is executed. If the first if is true, the block of code is executed, and the second if statement is tested. This checks to see whether the first number modulo the second number yields no remainder. If so, the numbers are either evenly divisible or equal. The if statement checks for equality and displays the appropriate message either way..



```
c:\WINDOWS\system32\cmd.exe
rNestedIf\Debug>OperatorNestedIf
Enter two numbers.
First: 10
Second: 2

They are evenly divisible!

C:\Documents and Settings\peter\My Documents\Visual Studio 2008\Projects\OperatorNestedIf\Debug>
```

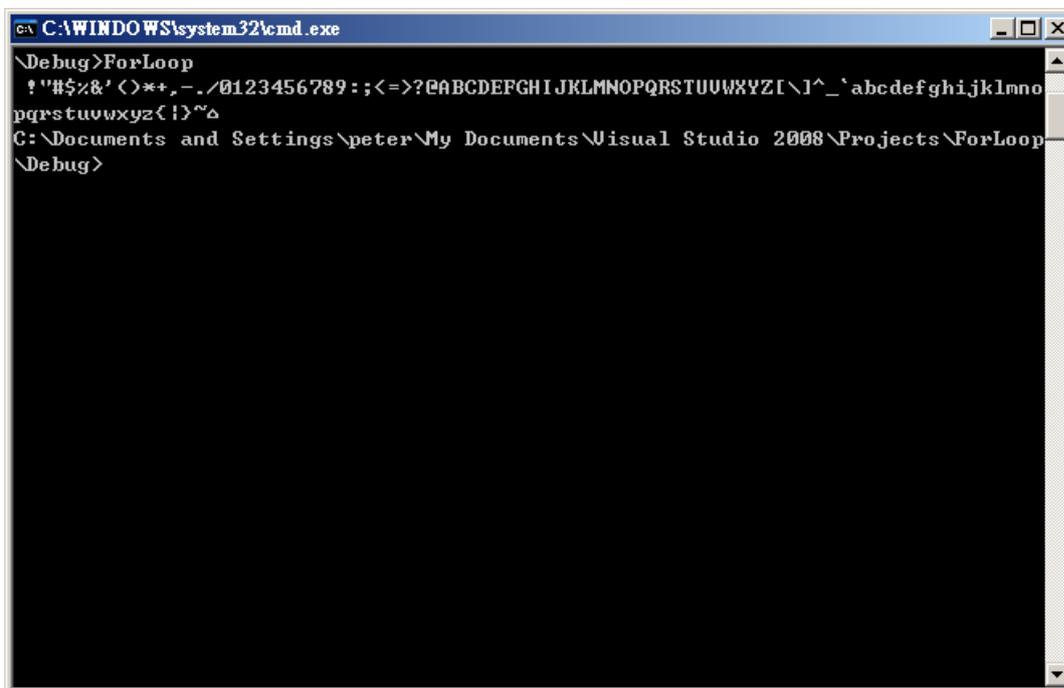
4. For Loop

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **ForLoop**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create a **C++ File (.cpp)** and name it as **main.cpp** by selecting **Project** → **Add New Item**, and then type the following code inside.

```
// Printing characters based on numbers.
#include <iostream>
using namespace std;

void main()
{
    for (int i = 32; i<128; i++)
        cout << (char) i;
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it.



```
C:\WINDOWS\system32\cmd.exe
\Debug>ForLoop
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ ^
C:\Documents and Settings\peter\My Documents\Visual Studio 2008\Projects\ForLoop
\Debug>
```

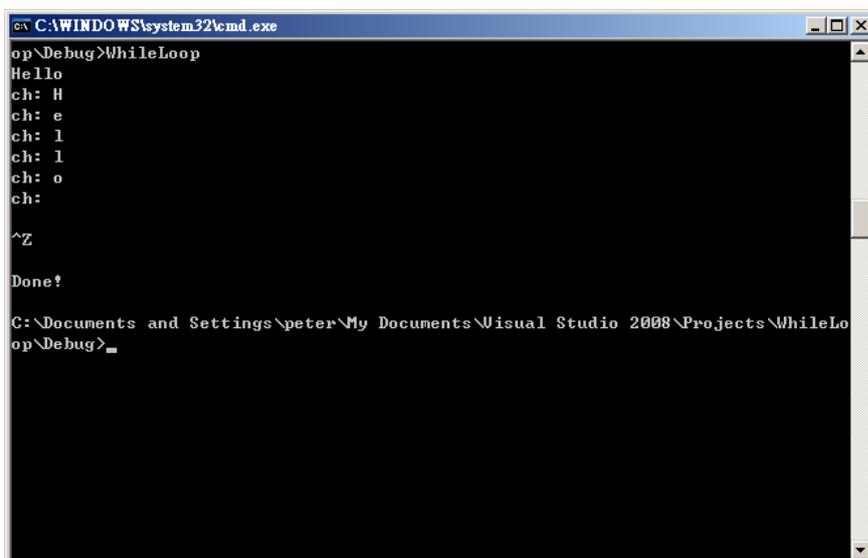
5. While Loop

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **WhileLoop**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create a **C++ File (.cpp)** and name it as **main.cpp** by selecting **Project** → **Add New Item**, and then type the following code inside.

```
// Using get() with no parameters
#include <iostream>
using namespace std;

void main()
{
    char ch;
    while ( (ch = cin.get()) != EOF)
    {
        cout << "ch: " << ch << endl;
    }
    cout << "\nDone!\n";
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it. The while loop assigns the input received from `cin.get()` to `ch`, and if it is not EOF the string is printed out. This output is buffered until an end of line is read, however. Once EOF is encountered (by pressing **[Ctrl] + [Z]** on a DOS machine, or **[Ctrl] + [D]** on a UNIX machine), the loop exits.



```
op\Debug>WhileLoop
Hello
ch: H
ch: e
ch: l
ch: l
ch: o
ch:
^Z
Done!
C:\Documents and Settings\peter\My Documents\Visual Studio 2008\Projects\WhileLo
op\Debug>
```

6. Using Getline

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **GetLine**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create a **C++ File (.cpp)** and name it as **main.cpp** by selecting **Project → Add New Item**, and then type the following code inside.

```
// Using getline()
#include <iostream>
using namespace std;

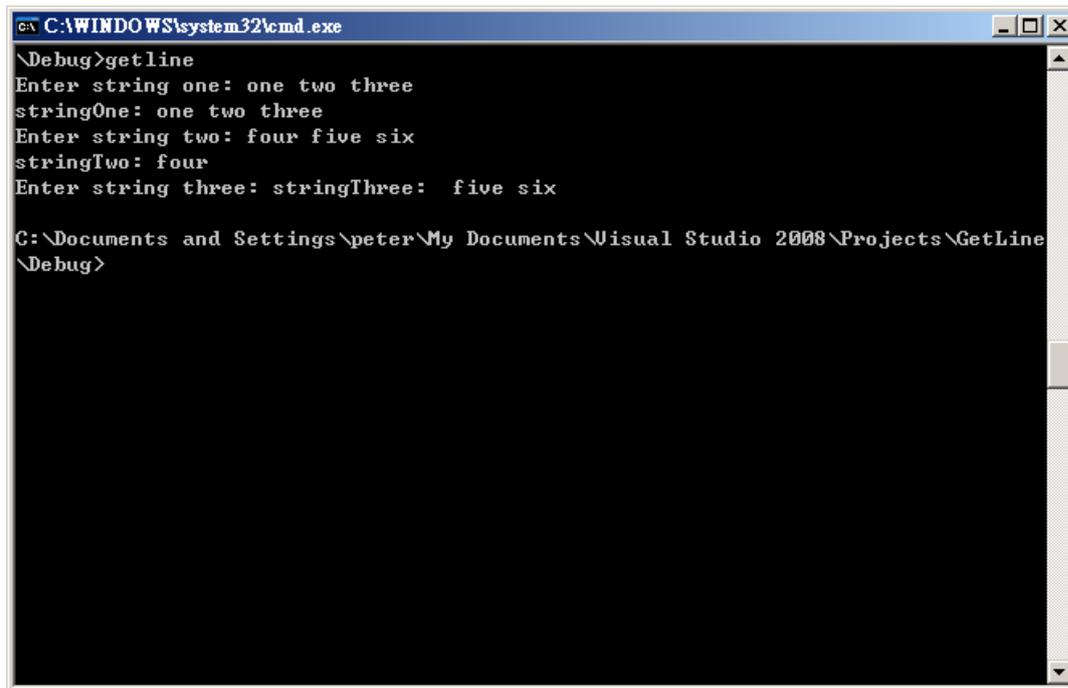
void main()
{
    char stringOne[256];
    char stringTwo[256];
    char stringThree[256];

    cout << "Enter string one: ";
    cin.getline(stringOne,256);
    cout << "stringOne: " << stringOne << endl;

    cout << "Enter string two: ";
    cin >> stringTwo;
    cout << "stringTwo: " << stringTwo << endl;

    cout << "Enter string three: ";
    cin.getline(stringThree,256);
    cout << "stringThree: " << stringThree << endl;
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it.



```
c:\WINDOWS\system32\cmd.exe
\nDebug>getline
Enter string one: one two three
stringOne: one two three
Enter string two: four five six
stringTwo: four
Enter string three: stringThree: five six

C:\Documents and Settings\peter\My Documents\Visual Studio 2008\Projects\GetLine
\nDebug>
```