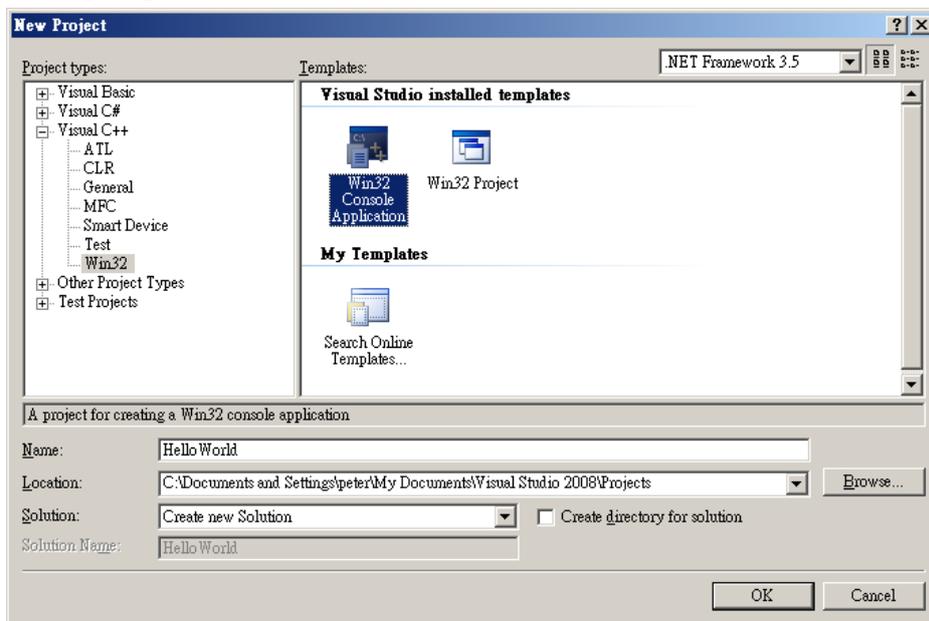
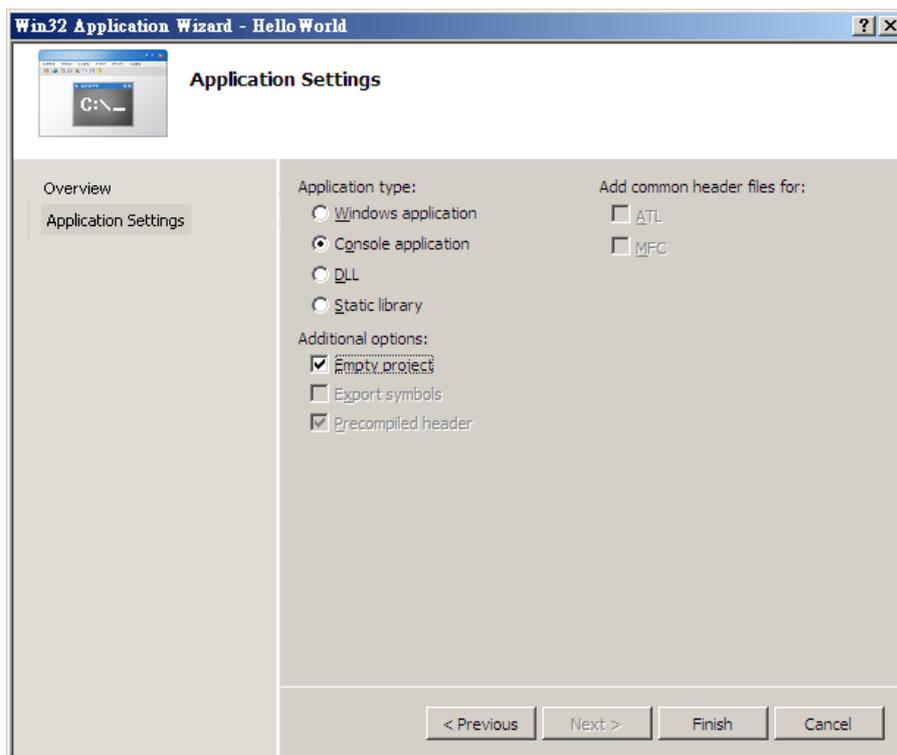


1. The First Visual C++ Program

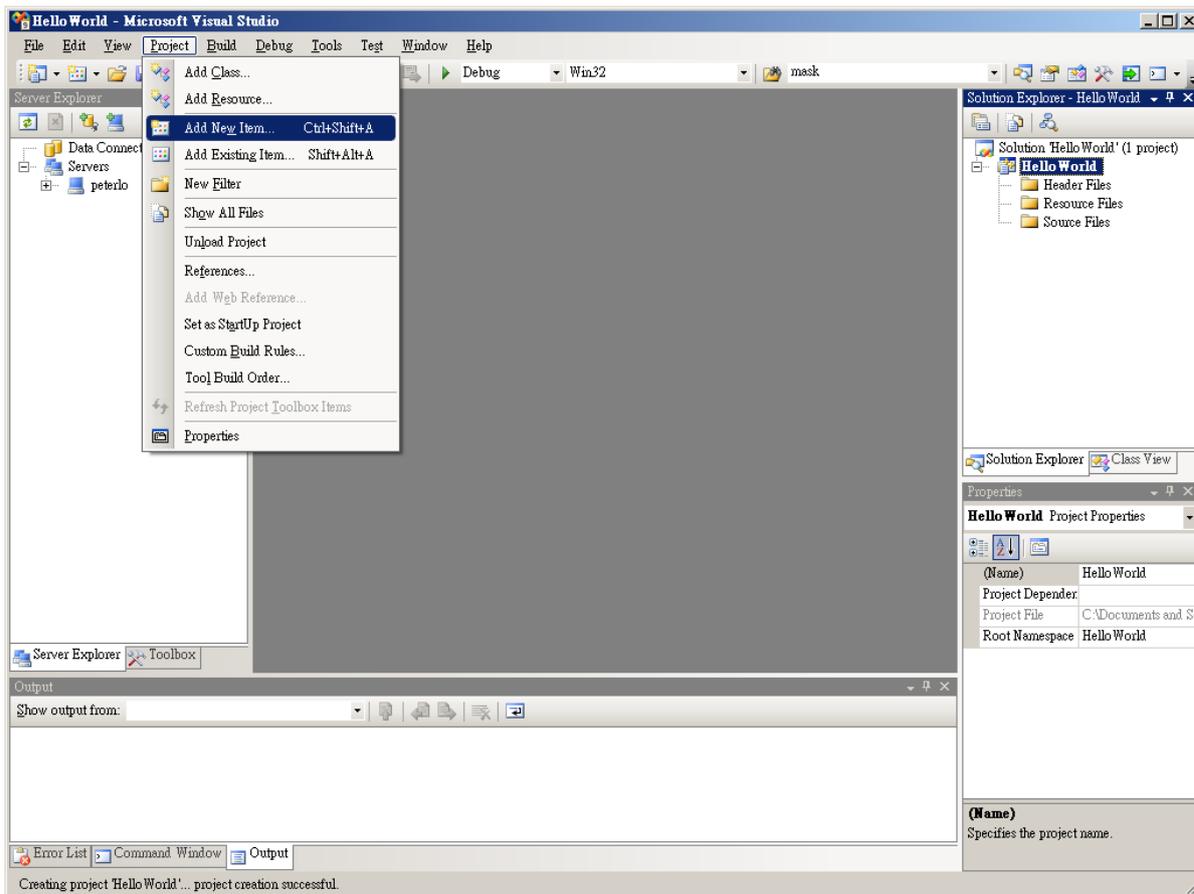
1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **HelloWorld**, and unselect the **Create directory for solution**. Press **[OK]** button to confirm.



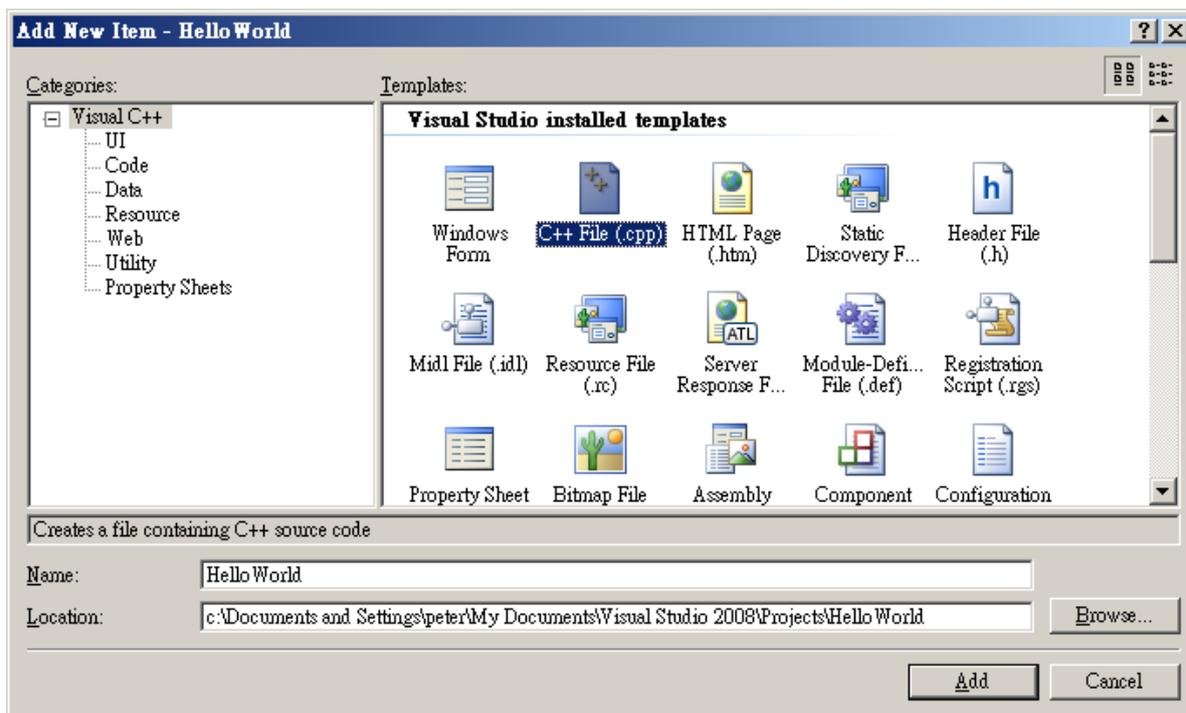
2. Select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**. Press **[Finish]** to continue.



3. In order to create the C++ file, select **Project** → **Add New Item**.



4. Select **C++ File (.cpp)** and name it as **HelloWorld.cpp**, and then press [Add] button to continue.



5. A blank screen will appear showing the contents of "HelloWorld". Type the following code into this blank white area.

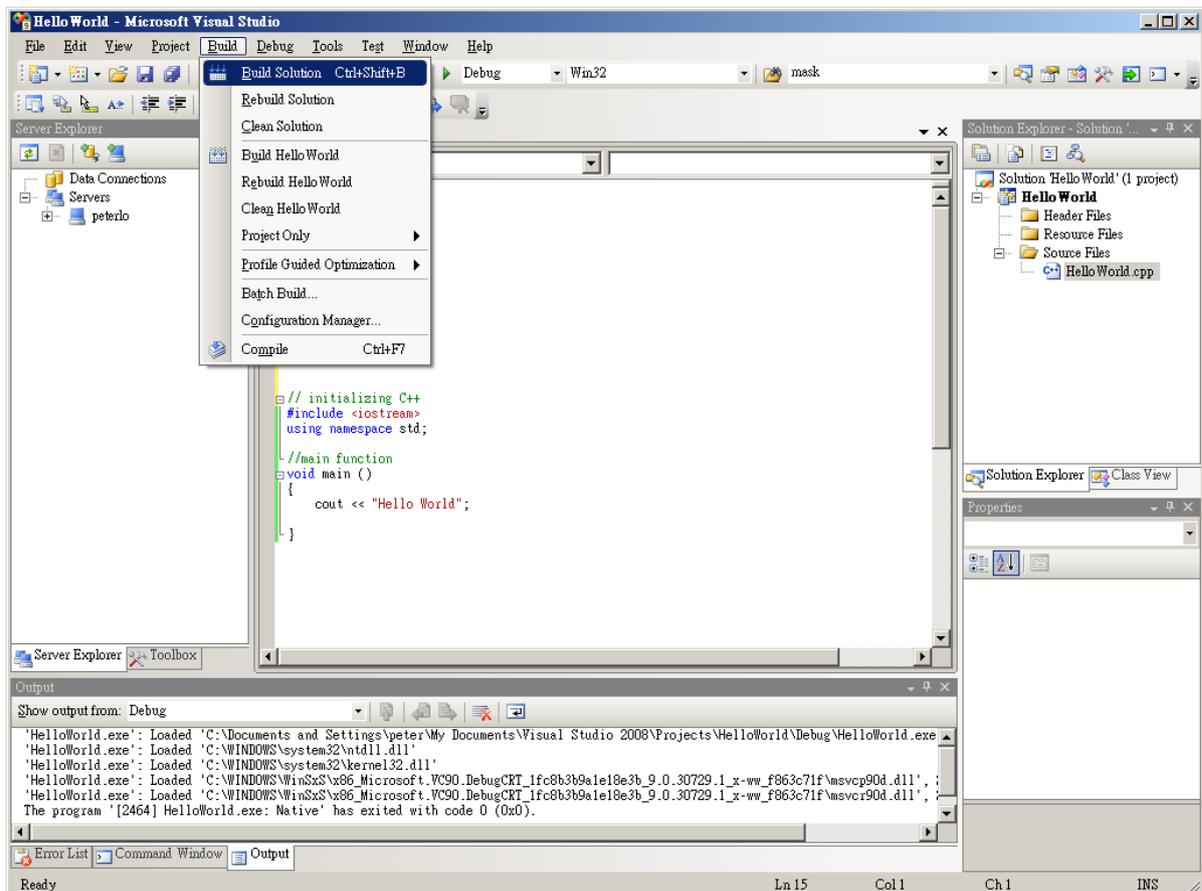
```
// initializing C++
#include <iostream>
using namespace std;

//main function
void main ()
{
    // Declare a variable
    char ch;

    // Print the statement to screen
    cout << "Hello World!! Press any key to continue:";

    // Capture the user input
    cin >> ch;
}
```

6. Compile the program by selecting **Build** → **Build Solution**.



7. Select **Start** → **Run** in your Windows, then type **CMD.EXE** to call up the DOS prompt. Then use command **CD** to go to **\My Documents\Visual Studio 2008\Projects\HelloWorld\Debug** and execute the command **HelloWorld.exe**.

On the other hand, you can also select **Debug** → **Start Debugging** to execute the program. The statement "Hello World!! Press any key to continue:" is displayed in the console, and you can press **[X]** with the **[Enter]** button to quit.



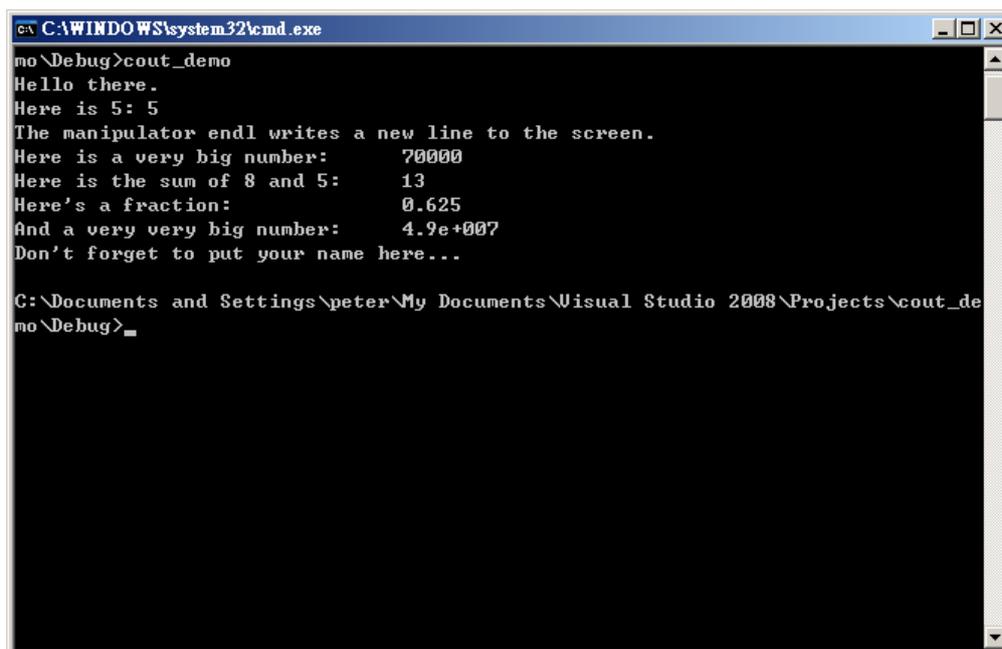
2. Output Stream using cout

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **cout_demo**. 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 **cout_demo.cpp** by selecting **Project → Add New Item**, and then type the following code inside.

```
// Program demonstrating the use of cout.
#include <iostream>
using namespace std;

void main()
{
    cout << "Hello there.\n";
    cout << "Here is 5: " << 5 << "\n";
    cout << "The manipulator endl writes a new line to the screen." << endl;
    cout << "Here is a very big number:\t" << 70000 << endl;
    cout << "Here is the sum of 8 and 5:\t" << 8+5 << endl;
    cout << "Here's a fraction:\t\t" << (float) 5/8 << endl;
    cout << "And a very very big number:\t" << (double) 7000 * 7000 << endl;
    cout << "Don't forget to put your name here...\n";
}
```

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



```
C:\WINDOWS\system32\cmd.exe
mo \Debug>cout_demo
Hello there.
Here is 5: 5
The manipulator endl writes a new line to the screen.
Here is a very big number:      70000
Here is the sum of 8 and 5:    13
Here's a fraction:             0.625
And a very very big number:    4.9e+007
Don't forget to put your name here...

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

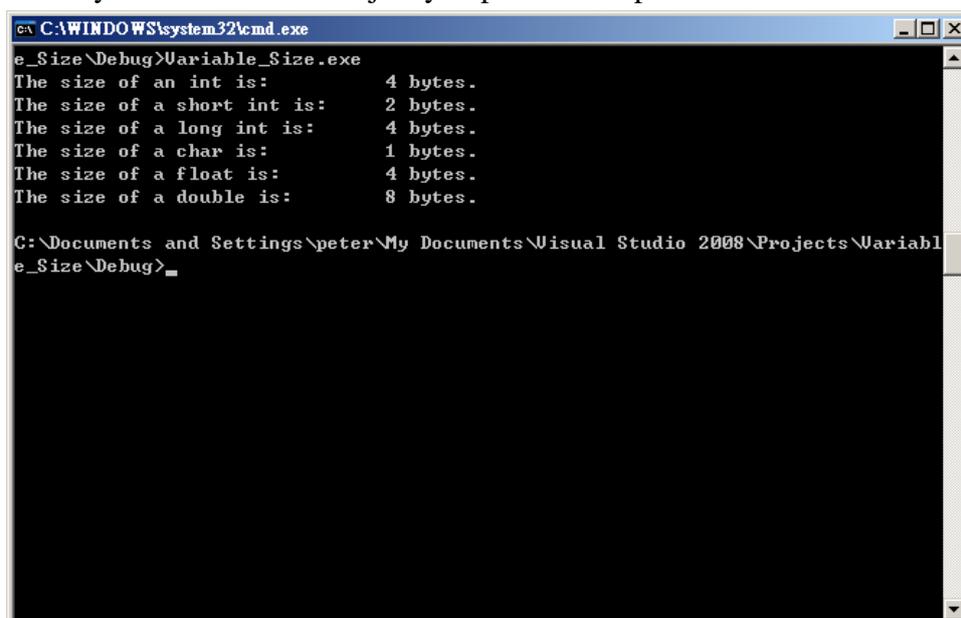
3. Variable Size

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **Variable_Size**. 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 program to determine the size of variable types on your computer.
#include <iostream>
using namespace std;

void main() {
    cout << "The size of an int is:\t\t"    << sizeof(int)    << " bytes.\n";
    cout << "The size of a short int is:\t" << sizeof(short) << " bytes.\n";
    cout << "The size of a long int is:\t"  << sizeof(long)  << " bytes.\n";
    cout << "The size of a char is:\t\t"   << sizeof(char)  << " bytes.\n";
    cout << "The size of a float is:\t\t"  << sizeof(float) << " bytes.\n";
    cout << "The size of a double is:\t"   << sizeof(double) << " bytes.\n";
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it. The one feature is the use of the `sizeof()` function in lines 6 through 11. `sizeof()` is provided by your compiler, and it tells you the size of the object you pass in as a parameter.



```

C:\WINDOWS\system32\cmd.exe
e_Size\Debug>Variable_Size.exe
The size of an int is:          4 bytes.
The size of a short int is:     2 bytes.
The size of a long int is:      4 bytes.
The size of a char is:         1 bytes.
The size of a float is:        4 bytes.
The size of a double is:       8 bytes.

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

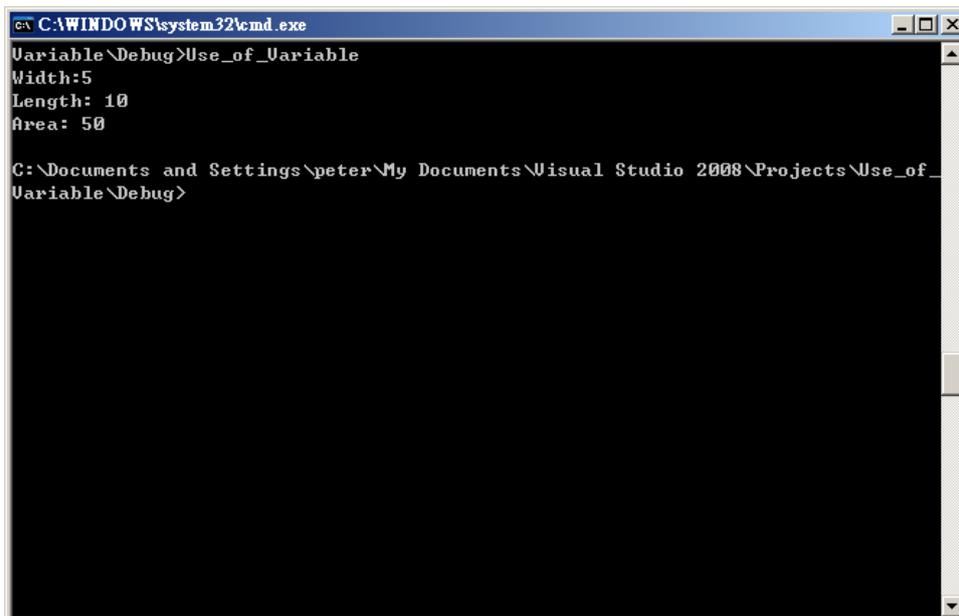
4. Use of Variables

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **Use_of_Variable**. 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.

```
// Demonstration of use variables
#include <iostream>
using namespace std;

void main()
{
    unsigned short int Width = 5, Length;
    Length = 10;
    // create an unsigned short and initialize with result
    // of multiplying Width by Length
    unsigned short int Area = Width * Length;
    cout << "Width:" << Width << "\n";
    cout << "Length: " << Length << endl;
    cout << "Area: " << Area << endl;
}
```

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



```
cmd C:\WINDOWS\system32\cmd.exe
Variable\Debug>Use_of_Variable
Width:5
Length: 10
Area: 50

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

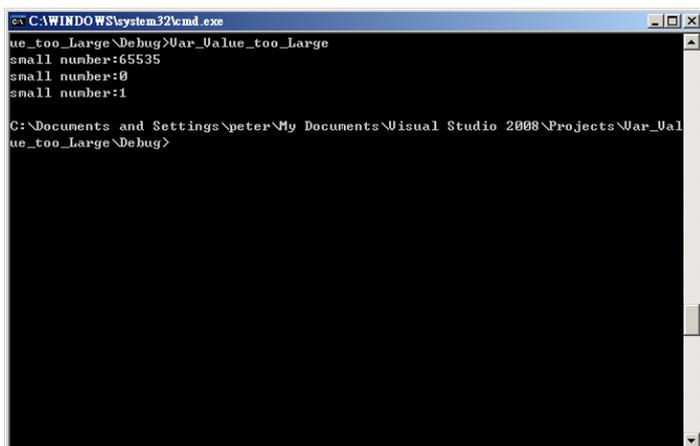
5. Demonstration of Overflow for Unsigned Integer

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **Var_Value_too_Large**. 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 putting too large a value in an unsigned integer.
#include <iostream>
using namespace std;

void main()
{
    unsigned short int small_number;
    small_number = 65535;
    cout << "small number:" << small_number << endl;
    small_number++;
    cout << "small number:" << small_number << endl;
    small_number++;
    cout << "small number:" << small_number << endl;
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it. Please note that the *small_number* is declared to be an *unsigned short int*, which on my computer is a two-bytes (it varies on different computer) variable, able to hold a value between 0 and 65,535. Then the maximum value is assigned to *small_number*, and it is printed on screen.



```
C:\WINDOWS\system32\cmd.exe
ue_too_Large\Debug>Var_Value_too_Large
small number:65535
small number:0
small number:1

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

6. Demonstrates Subtraction and Integer Overflow

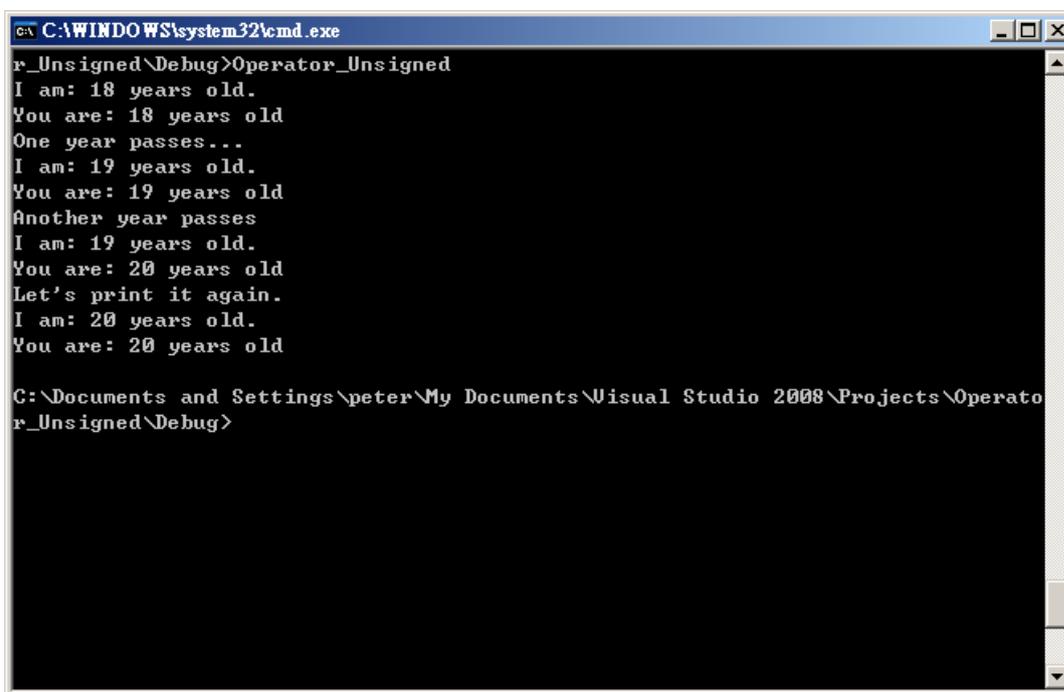
1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **Operator_Unsigned**. Remember to select **Application Setting** in the **Win32 Application Wizard**, then select **Console Application** and select **Empty Project**.
2. Create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

```
// demonstrates use of prefix and postfix increment and decrement operators
#include <iostream>
using namespace std;

void main()
{
    int myage = 18;        // initialize two integers
    int yourage = 18;
    cout << "I am: " << myage << " years old.\n";
    cout << "You are: " << yourage << " years old\n";
    myage++;              // postfix increment
    ++yourage;           // prefix increment

    cout << "One year passes...\n";
    cout << "I am: " << myage << " years old.\n";
    cout << "You are: " << yourage << " years old\n";
    cout << "Another year passes\n";
    cout << "I am: " << myage++ << " years old.\n";
    cout << "You are: " << ++yourage << " years old\n";
    cout << "Let's print it again.\n";
    cout << "I am: " << myage << " years old.\n";
    cout << "You are: " << yourage << " years old\n";
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it.
- Please note that two integer variables are declared, and each is initialized with the value 18. Their values are printed on lines 9 and 10. On line 11, *myage* is incremented using the postfix increment operator, and on line 12, *yourage* is incremented using the prefix increment operator. The results are printed on lines 14 and 15, and they are identical (both 19).
 - *myage* is incremented as part of the printing statement, using the postfix increment operator. Because it is postfix, the increment happens after the print, and so the value 19 is printed again. In contrast, on line 18, *yourage* is incremented using the prefix increment operator. Thus, it is incremented before being printed, and the value displays as 20.
 - Finally, the values are printed again. Because the increment statement has completed, the value in *myage* is now 20, as is the value in *yourage*..



```
c:\C:\WINDOWS\system32\cmd.exe
r_Unsigned\Debug>Operator_Unsigned
I am: 18 years old.
You are: 18 years old
One year passes...
I am: 19 years old.
You are: 19 years old
Another year passes
I am: 19 years old.
You are: 20 years old
Let's print it again.
I am: 20 years old.
You are: 20 years old

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

7. Compile Errors

1. Start the Microsoft Visual Studio and start a new Visual C++ Project. Select **Win32 Console Application** and name it as **Error**. 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.

```
// Error.cpp
#include <iostream>
using namespace std;

void main ()
{
    int i;
    float j=0.9           // a missing ";"

    i=j;                 // only a warning here, note that the value of i
                        // will become 0 since the .9 will be truncated

    cout << "test" << endl;
    cout << "test 2 i: << i << endl; // a missing double quote
                        // a missing ";"
}
```

3. Use **Build Solution** to compile and build the solution, and then execute it. Can you successfully execute the program?

