

1. File Handling

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

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

void main()
{
    char fileName[80];
    char buffer[255];           // for user input

    cout << "File name: ";
    cin >> fileName;

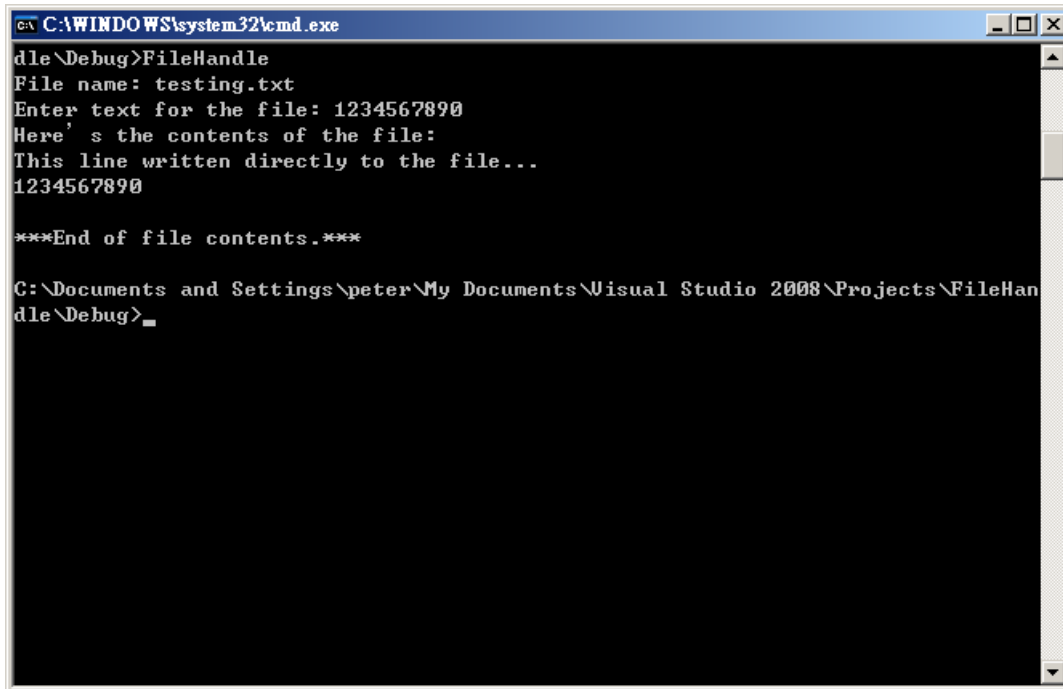
    ofstream fout(fileName);   // open for writing
    fout << "This line written directly to the file...\n";
    cout << "Enter text for the file: ";
    cin.ignore(1, '\n');       // eat the newline after the file name
    cin.getline(buffer,255);    // get the user's input
    fout << buffer << "\n";    // and write it to the file
    fout.close();              // close the file, ready for reopen

    ifstream fin(fileName);    // reopen for reading
    cout << "Here's the contents of the file:\n";
    char ch;
    while (fin.get(ch))
        cout << ch;

    cout << "\n***End of file contents.***\n";

    fin.close();               // Close the file
}
```

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

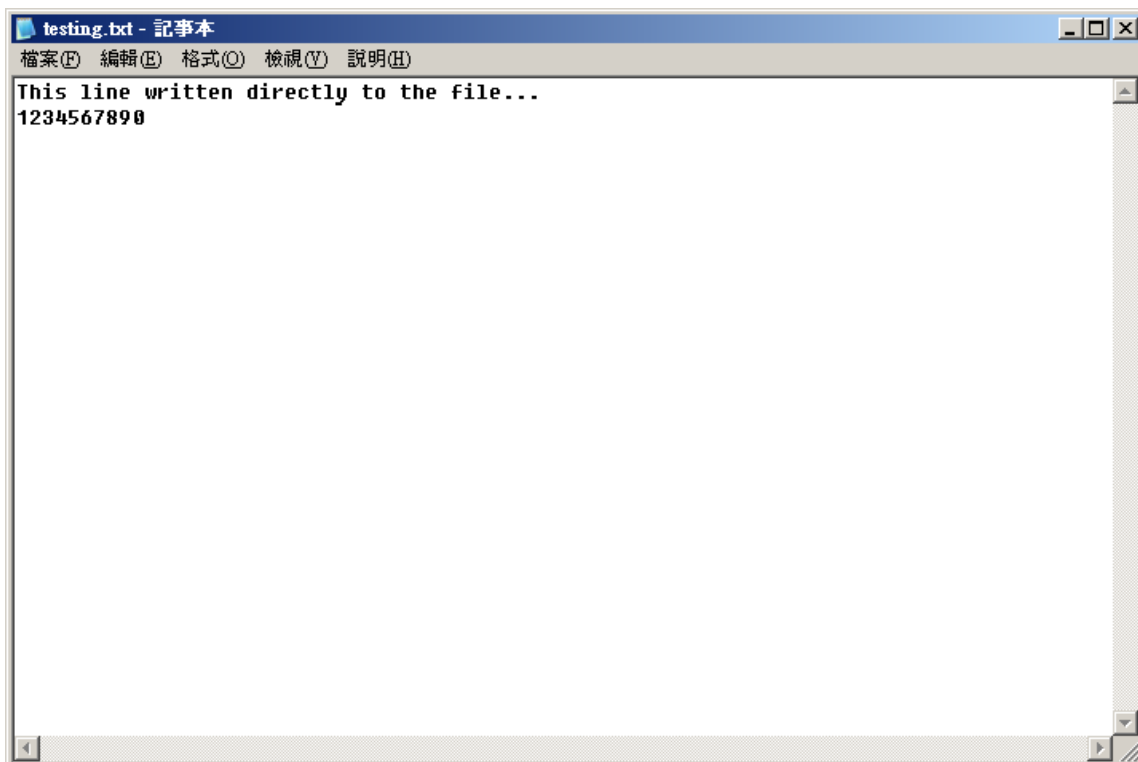


```
c:\WINDOWS\system32\cmd.exe
d:\Debug>FileHandle
File name: testing.txt
Enter text for the file: 1234567890
Here's the contents of the file:
This line written directly to the file...
1234567890

***End of file contents.***

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

4. Use **Notepad** to open the text file you created, is the file content the same?



```
testing.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
This line written directly to the file...
1234567890
```

2. Save to File

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

```
// save to a file
#include <iostream>
#include <fstream>
using namespace std;

void main()
{
    char filename[20];
    char ch;

    cout << "Enter name for new file: ";
    cin >>filename;

    // create output stream object for new file and call it fout
    ofstream fout(filename);
    fout << "For your eyes only!\n";      // write to file
    cout << "Enter your secret number: "; // write to screen

    float secret;

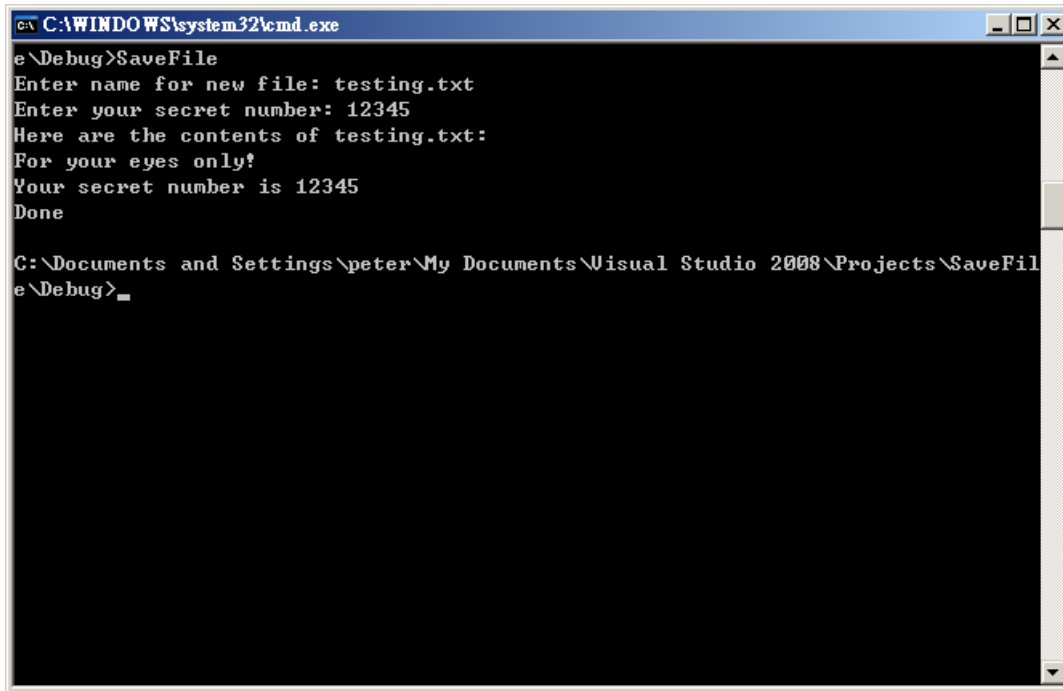
    cin >>secret;
    fout << "Your secret number is " << secret << "\n";
    fout.close();          // close file

    // create input stream object for new file and call it fin
    ifstream fin(filename);
    cout << "Here are the contents of " << filename << ":\n";

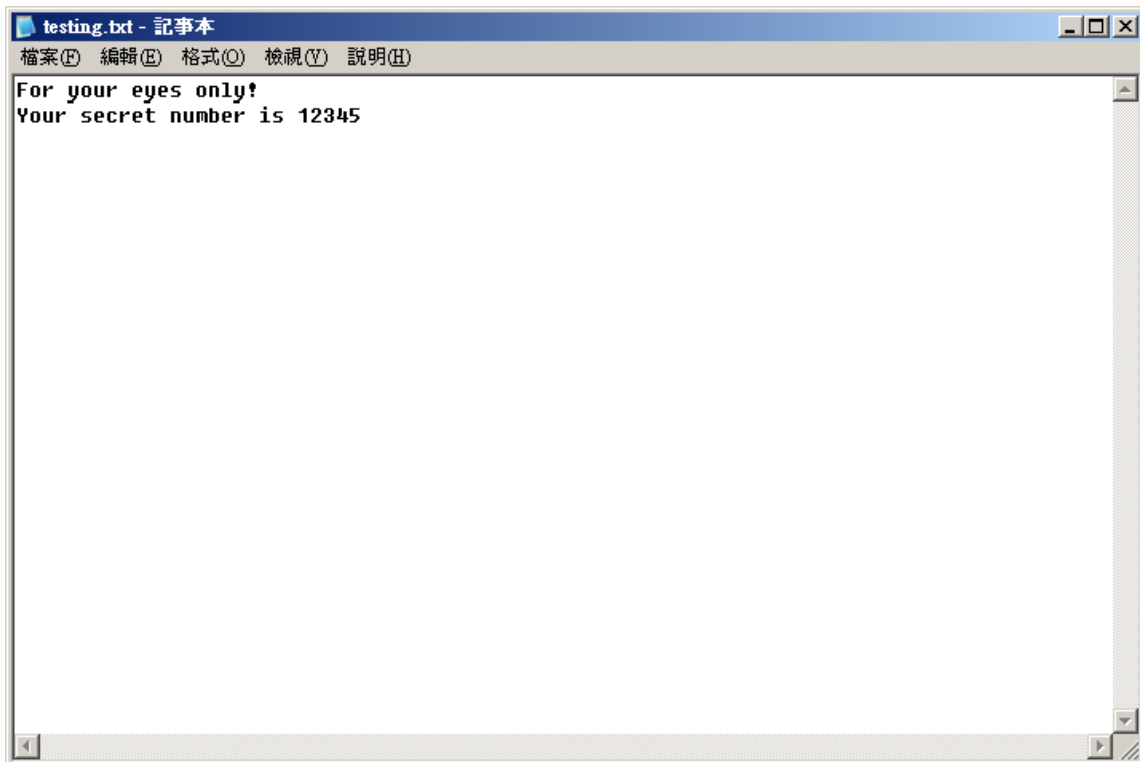
    while (fin.get(ch))    // read character from file and
        cout << ch;        // write it to screen
    cout << "Done\n";

    fin.close();          // Close the file
}
```

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



4. Use **Notepad** to open the text file you created, is the file content the same?



3. Simple Data Structure

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

```
// Purpose : An example of creating and using a simple data structure
#include <iostream>
#include <string>
using namespace std;

// Create a mailing address data structure
struct MailingAddress
{
    char Name[30];
    char Street[80];
    char District[30];
    char City[30];
};

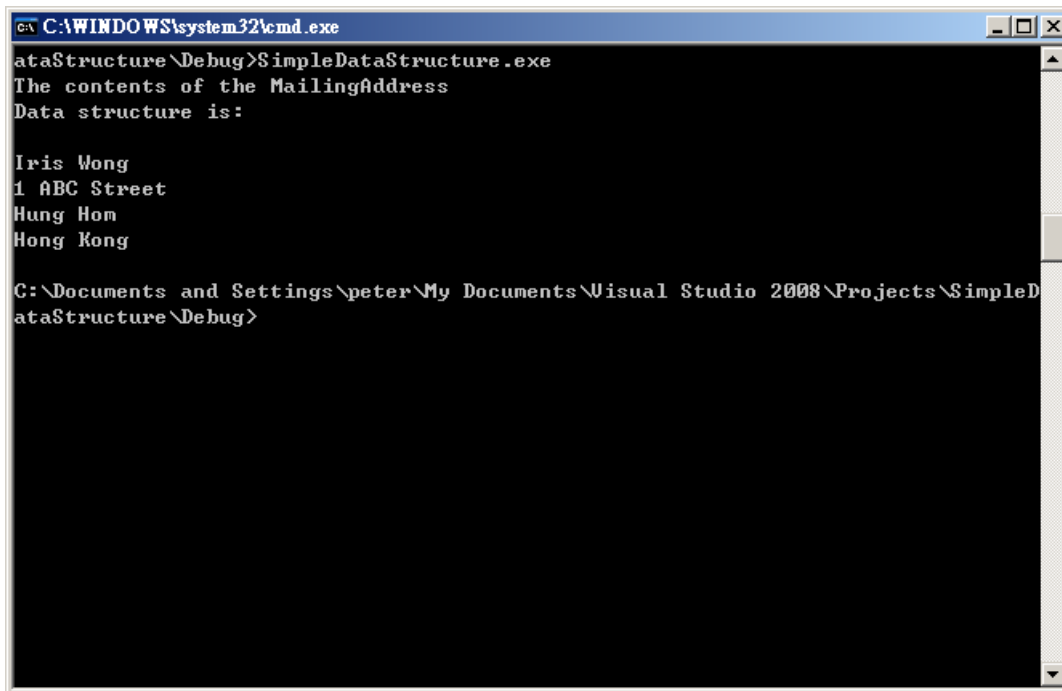
void main()
{
    // Declare a variable of type MailingAddress
    MailingAddress ma;

    // Populate the structure with some data
    strcpy(ma.Name, "Iris Wong" );
    strcpy(ma.Street, "1 ABC Street");
    strcpy(ma.District, "Hung Hom" );
    strcpy(ma.City, "Hong Kong" );

    // Access the data stored in the structure
    cout << "The contents of the MailingAddress \n" << "Data structure is:\n\n";

    cout << ma.Name << endl;
    cout << ma.Street << endl;
    cout << ma.District << endl;
    cout << ma.City << endl;
}
```

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



```
C:\WINDOWS\system32\cmd.exe
ataStructure\Debug>SimpleDataStructure.exe
The contents of the MailingAddress
Data structure is:

Iris Wong
1 ABC Street
Hung Hom
Hong Kong

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

4. Pointer Structure

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

```
////////////////////////////////////  
// PointerStructure  
//  
// Purpose : Demonstrates several ways to access a  
//           structure member using the indirection  
//           and dot operators.  
////////////////////////////////////  
#include <iostream>  
using namespace std;  
  
const char NL = '\n';  
  
// The Image structure  
//  
typedef struct  
{  
    int    width;  
    int    height;  
    unsigned *bits;  
}  
Image;  
  
////////////////////////////////////  
// main()  
  
void main()  
{  
    Image Picture1;  
    Image *pImage = &Picture1;  
  
    Picture1.width = 100;  
  
    cout << "Picture1.width    == "  
    << Picture1.width << NL;
```

```

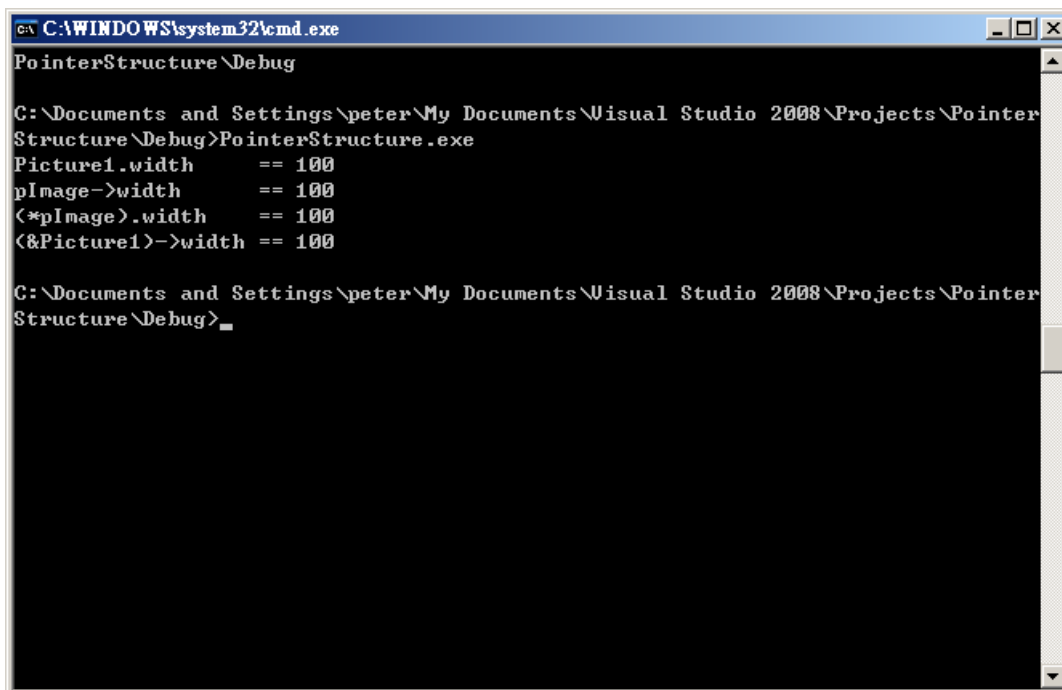
cout << "pImage->width    == "
<< pImage->width << NL;

cout << "(*pImage).width  == "
<< (*pImage).width << NL;

cout << "(&Picture1)->width == "
<< (&Picture1)->width << NL;
}

```

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

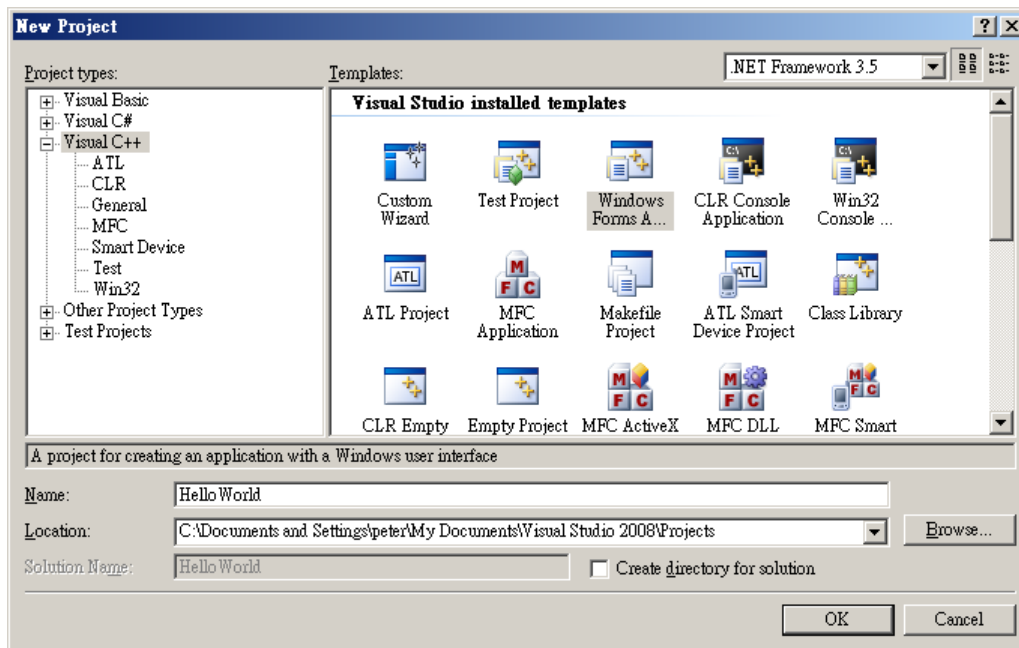


4. Program Explanation

- Lines 15-21: Declare a structure called Image
- Lines 28-29: Create an Image object and initialize a pointer to an Image object
- This program uses the Image structure declared from lines 15 through 21 to demonstrate data member access using a pointer. The first task of the main() function is to create an instance of an Image structure (Picture1) on line 28. Line 29 defines a pointer to an Image object (pImage) and initializes it with the address of the Picture1 object (&Picture1).
- On line 31, the value of the Picture1.width member is initialized to the value 100. The remaining statements in the main() function show various ways of accessing the width member. Line 34 uses the Picture1.width member directly.
- Line 37 uses the pointer pImage to access the width member, using the pointer-to-member operator (->).
- Line 40 uses the dereferenced pImage pointer (*pImage) with the dot operator (.) to access the width member.
- Line 43 uses the address of the Picture1 object (&Picture1) with the pointer-to-member operator to access the width member.

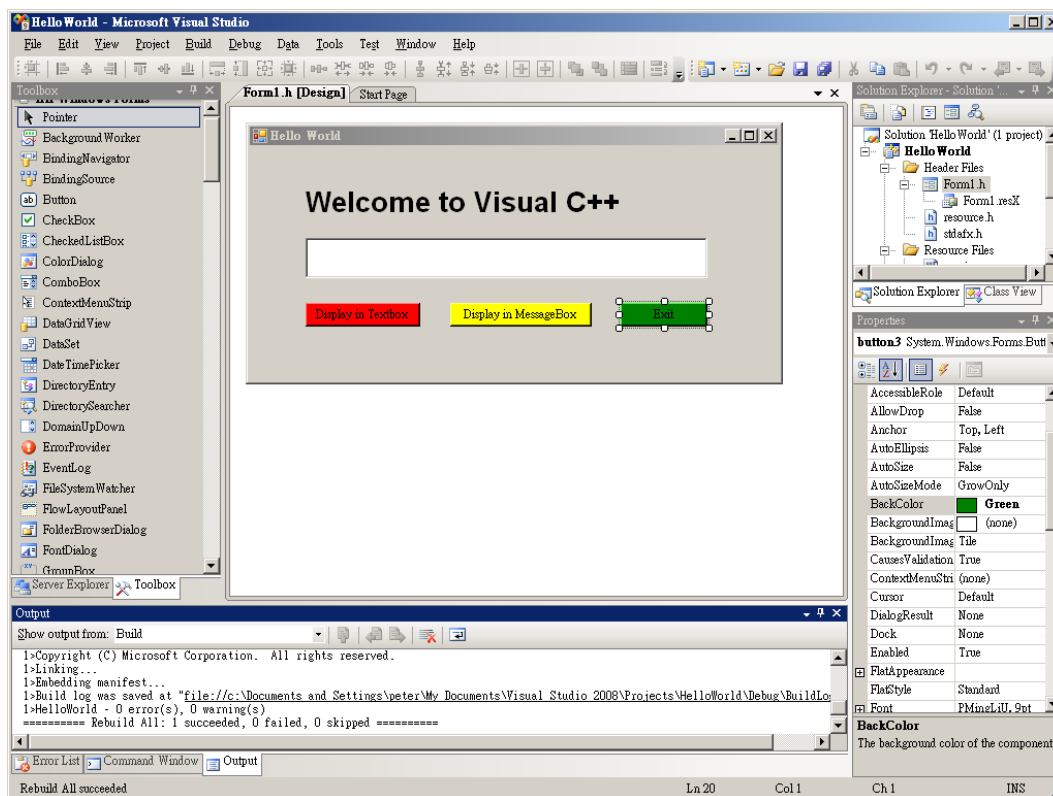
5. Windows Form Programming

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



2. From the Toolbox, drag a Label, a TextBox and three Button controls onto the form and customize the properties.

Object	Name	Property	Property Value
Form	Form1	Text	Hello World
Label	label1	Text	Welcome to Visual C++
		Font Name	Arial
		Font Size	20 point
		Font Style	Bold
TextBox	textBox1	Text	(Blank)
Button	button1	Text	Display in Text Box
		BackColor	Red
	button2	Text	Display in Message Box
		BackColor	Yellow
	button3	Text	Exit
		BackColor	Green



3. Double click the “Display in TextBox” button (**button1**) to enter “Code Edit” mode. In the **Click** event procedure of the **button1** control, add the following code.

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->textBox1->Text = "Hello World";
}
```

4. Double click the “Display in Display in MessageBox” button (**button2**) to enter “Code Edit” mode. In the **Click** event procedure of the **button2** control, add the following code.

```
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    MessageBox::Show("Hello World!", "Title", MessageBoxButtons::OK,
        MessageBoxIcon::Asterisk);
}
```

5. Double click the “Exit” button (**button3**) to enter “Code Edit” mode. In the **Click** event procedure of the **button3** control, add the following code.

```
private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->Close();
}
```

6. Select **Build** → **Rebuild Solution** to compile and then press **Start** → **Start Debugging** to execute the program. Can you observe the different by pressing different button?

