CHAPTER 1

Hello World

Choosing an IDE

To begin developing in C++ you should download and install an Integrated Development Environment (IDE) that supports C++. A good choice is Microsoft's own Visual Studio.[1] If you do not have Visual Studio but would like to try out the examples in this book in a similar environment you can download Visual Studio Express[2] from Microsoft's website. This is a lightweight version of Visual Studio that is available for free. Alternatively, you can develop using a simple text editor – such as Notepad – although this is less convenient than using an IDE. If you choose to do so, just create an empty document with a .cpp file extension and open it in the editor of your choice.

Creating a project

After installing Visual Studio or Visual Studio Express, go ahead and launch the program. You then need to create a project, which will manage the C++ source files and other resources. Go to File ➤ New ➤ Project in Visual Studio, or File ➤ New Project in Visual Studio Express, to display the New Project window. From there select the Visual C++ template type in the left frame. Then select the Win32 Console Application template in the right frame. At the bottom of the window you can configure the name and location of the project. When you are finished, click the OK button and another dialog box will appear titled Win32 Application Wizard. Click next and a couple of application settings will be displayed. Leave the application type as Console application and check the Empty project checkbox. Then click Finish to let the wizard create your empty project.

Adding a source file

You have now created a C++ project. In the Solution Explorer pane (View ➤ Solution Explorer) you can see that the project consists of three empty folders: Header Files, Resource Files and Source Files. Right click on the Source Files folder and select Add ➤ New Item. From the Add New Item dialog box choose the C++ File (.cpp) template. Give this source file the name "MyApp" and click the Add button. An empty cpp file will now be added to your project and also opened for you.

[1] http://www.microsoft.com/visualstudio
[2] http://www.microsoft.com/express

Hello world

The first thing to add to the source file is the main function. This is the entry point of the program, and the code inside of the curly brackets is what will be executed when the program runs. The brackets, along with their content, is referred to as a code block, or just a block.

```
int main() {}
```

The first application will simply output the text "Hello World" to the screen. Before this can be done the iostream header needs to be included. This header provides input and output functionality for the program, and is one of the standard libraries that come with all C++ compilers. What the `#include` directive does is effectively to replace the line with everything in the specified header before the file is compiled into an executable.

```
#include <iostream>
int main() {}
```

With iostream included you gain access to several new functions. These are all located in the standard namespace called `std`, which you can examine by using a double colon, also called the scope resolution operator (`::`). After typing this in Visual Studio, the IntelliSense window will automatically open, displaying what the namespace contains. Among the members you find the `cout` stream, which is the standard output stream in C++ that will be used to print text to a console window. It uses two less-than signs known as the insertion operator (`<<`) to indicate what to output. The string can then be specified, delimited by double quotes, and followed by a semicolon. The semicolon is used in C++ to mark the end of all statements.

```
#include <iostream>

int main()
{
  std::cout << "Hello World";
}
```

Using namespace

To make things a bit easier you can add a line specifying that the code file uses the standard namespace. You then no longer have to prefix `cout` with the namespace (`std::`) since it is now used by default.

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

int main()
{
  cout << "Hello World";
}
```