Chapter 1 – Introduction

Hi! Welcome to my beginners guide to C++. If you are starting to program for the first time, I hope that you find the chapters I have written useful. C++ is an excellent language to start programming in – a lot of applications that you use are probably written in c++ and once you learn some basic concepts, learning other languages, like java for example, will be much easier.

There are 8 chapters altogether, including this one. I have kept the chapters short and concise so you won't get bored or weighed down by too much information. After each chapter you can rearrange the code in the examples provided or make up your own code. Programming is very much a practical subject so you will learn a lot by messing about with code or even looking at other people's code.

What is a computer program?

A computer program is a set of instructions that a programmer writes to tell a computer how to carry out a certain task. The instructions, however, must be in a language that the computer understands. Computers only understand binary language i.e. that composed of 1's and 0's. This is a low level language and very hard to program in. So humans invented higher level languages such as C++ or Pascal to make the job easier. As you will see, these languages are nearly like English but you don't have the freedom to write what you like – there are still rules you have to follow.

To convert a program in C++ to a binary or executable file that the computer can understand we use a *compiler*. The compiler which I recommend using for this guide is Borland's command line compiler. This compiler is completely free to use and can be downloaded from:

http://www.borland.com/products/downloads/download_cbuilder.html Or you could do a search on google for a free Borland compiler.

Setting up Borland's command line compiler

Now I will show you how to set up the Borland compiler as it can be a bit confusing if you are new to this stuff.

When you download the Borland compiler you will see a file like the following:

freecommandLinetools
PackageForTheWeb Stub
InstallShield Software Corpora...

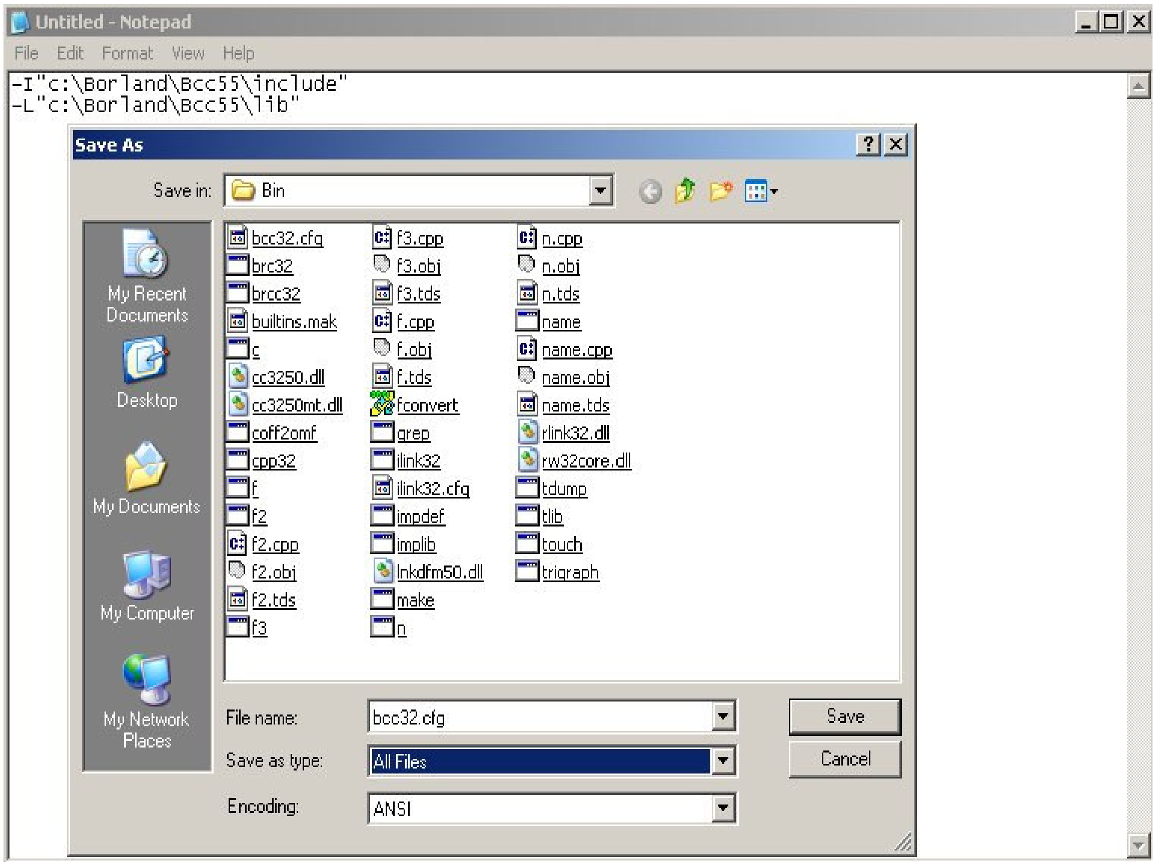
Step 1: Click on the file and install it to the default directory **C:\Borland\BCC55**

Step 2: Open notepad and type the following lines:

```
-I"c:\Borland\Bcc55\include"
-L"c:\Borland\Bcc55\lib"
```

Save this file in **C:\Borland\BCC55\BIN** as **bcc32.cfg**

Make sure you choose "All Files" when saving or it will be saved as a text file.



Untitled - Notepad

```
-I"c:\Borland\Bcc55\include"
-L"c:\Borland\Bcc55\lib"
```

Save As

Save in: Bin

File name: bcc32.cfg

Save as type: All Files

Encoding: ANSI

Step 3:

Open notepad again and type:

```
-L"c:\Borland\Bcc55\lib"
```

Save this file in **C:\Borland\BCC55\BIN** as **ilink32.cfg**

Again, make sure you choose “All Files” when saving or it will be saved as a text file.

That’s it! All the paths are set up and now you can get stuck into some programming!

Writing your first program in C++

With the Borland compiler, you type your program into a text editor such as notepad then you compile and run it from the command line.

So open notepad and type the following code into it.

```
#include<iostream.h>

int main()
{

cout<<"Hello World!"<<endl;

return 0;

}
```

Example 1a

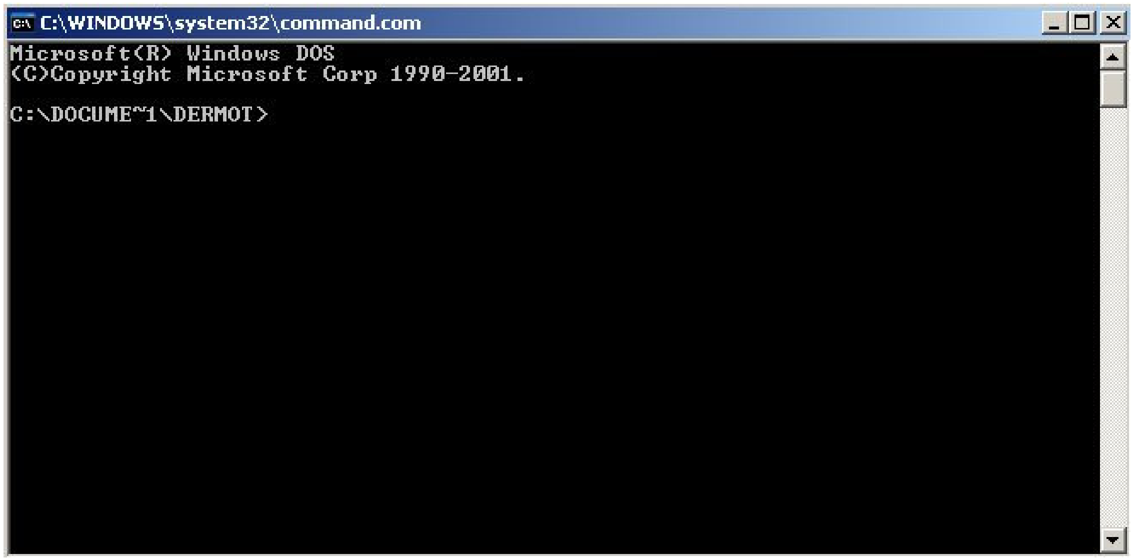
Save it in **C:\Borland\BCC55\BIN** as **p1.cpp**

Remember to choose “All files” when saving.

Compiling your first program in C++

Now your program is saved and we need to compile it. First we need to open up a command line.

Click on **Start**, and then **run**. In the box that comes up type “**command**” and a command line will pop up.



```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.

C:\DOCUME~1\DERMOT>
```

Navigate to the folder where your file is saved by typing “**cd c:\borland\bcc55\bin**” and press enter.

The name of the compiler is bcc32.exe so to compile a program all we need to do is type **bcc32** followed by the name of the program.

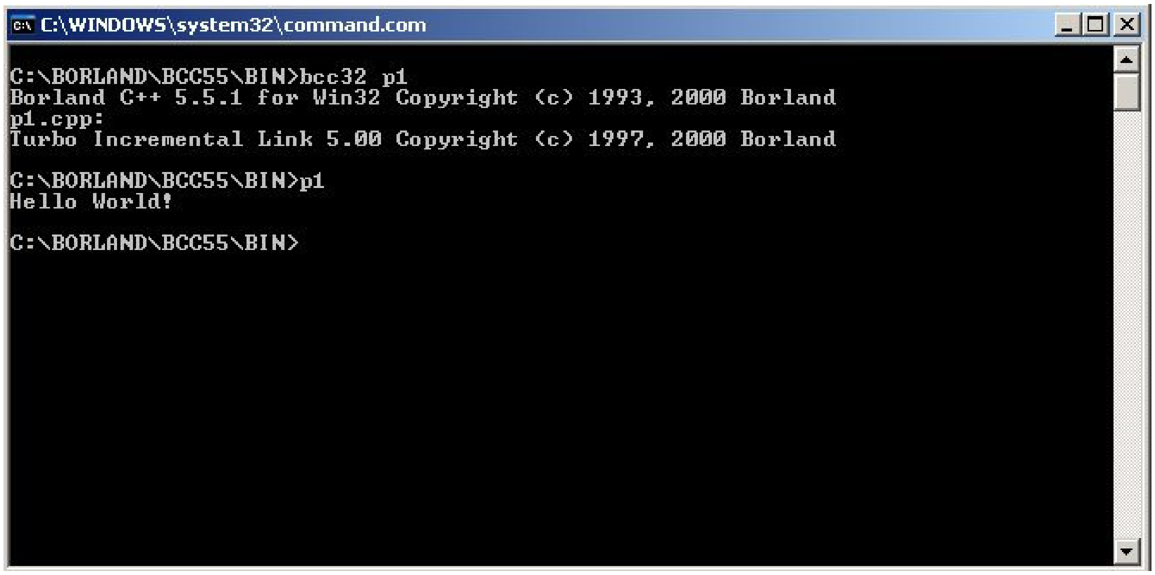
In our case we should type: **bcc32 p1** (you don’t have to type the .cpp bit here).

Now **press enter** and the program will be compiled.

You shouldn’t get any errors if you typed the program correctly but if you did you will be notified here.

To run the program just type **p1** and bingo! “Hello World!” is printed on the screen. Congratulations! You’ve just written and compiled your first program in C++!

To run the program again just type p1 followed by enter. You only need to re-compile the program if you make any changes.



```
C:\BORLAND\BCC55\BIN>bcc32 p1
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
p1.cpp:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

C:\BORLAND\BCC55\BIN>p1
Hello World!

C:\BORLAND\BCC55\BIN>
```

Understanding your first program

The first line tells the compiler to include a file called "iostrem.h". You will find this file in a folder called "include" in “C:\Borland\BCC55\. This file tells the compiler what certain keywords mean, such as "cout" on line 6.
For instance later on you will see that you will have to include <math.h> in order to use reserved or keywords like "sine" and "rad". You will probably start all of your programs with <iostream.h>

Line three, "int main()", is the header for the main function- where most of the programming stuff happens. Later on you will learn to write your own functions which can be called from the main function. Notice that the body of the function is enclosed in brackets, {....}. At the end of the function there is a line "return 0;". This tells the computer that when main() is finished what it is doing it returns or does nothing.

Line 6, cout<<"Hello World!"; is called a statement. All statements end with a semi colon. It is very easy to forget to put this in when you begin programming. "cout" tells the computer to print on the screen the text between the pair of quotes. cout stands for

"Channel OUT". Also notice the two pairs of arrows after cout and their direction. To remember their direction, think of data flowing "out" to be printed on the screen. "endl" tells the computer to go onto a new line after it prints the current one. This is useful if you want different sentences on different lines.
To put a space in between lines printed on the screen, type
cout<<" "<<endl; in between the code lines.

So now that you know how to write a program that prints something, you can edit the above program to output whatever you like. Repeat line 6 above to print out a few sentences in the same program.

As you read through the various sections of this guide let me know what you think. You can email me at dermot.mceneaney2@mail.dcu.ie