

Hello World Using Bloodshed Dev C++

By Janine Bouyssounouse

Bloodshed Dev C++ is a free program to make writing and compiling (doing the stuff in the background that lets the computer understand what is written in a program) C++ programs easy to do. It makes executable files (programs that run by double clicking on an icon) quickly so programs can be shared with others as soon as they are written and debugged (getting the mistakes out of the program).

Bloodshed Dev C++ can be downloaded from the website: <http://www.bloodshed.net/devcpp.html>. There is a link at the bottom of the overview of the program in the center of the screen. There are many resources available on this site to explain how to download the program and how to use it. Clicking on the Dev C++ section brings up a list of features and system requirements for the software. Find the latest full version of the software on the page and follow the downloading and installation instructions.

Once the software is downloaded and installed on the computer, then it is time to write the first program, compile it and turn it into an executable file. Hello World is the classic first program. It is very short and is up and running in no time flat. Going through the process makes sure that the software is installed and the basic functions are working. It also helps jump start the programming adventure by walking through the basic steps of software development. This topic gets much more complicated in the real world, but here it will be viewed in a simplified version.

There is more than one way to start writing programs in Dev C++. This document discusses using a project for each program. This encourages using an organized file structure to keep each program in a separate location for future use. Remember that reusing code is a good idea and it is nice to be able to find the folder with everything for a program in one place.

Click on the File menu and select New – Project. Choose Console Application for this first program. Type in a name for the project, such as HelloWorld, in the name field in the bottom left corner of the window. Click on the OK button. Choose a place on the computer to save the project. This can be on the hard drive, on a shared drive or on an external storage device.

Once the project is saved, some basic items show up on the screen. These items give a shell of a C++ program and include things that are unique to Bloodshed Dev C++ as well.

```
#include <iostream>    - This tells the compiler to include files.
#include <stdlib.h>    - This tells the compiler to include files.

using namespace std;  - This tells the compiler that you will be using key
                    words that are included in namespace std.

int main(int argc, char *argv[])    - This starts the main function.
{                                    - Main is contained between { and }.

    system("PAUSE");                - This leaves the console window open until
                                    you are ready to close it. Otherwise you
                                    wouldn't be able to see the results of the
                                    program. Not all compilers require this.

    return 0;                        - This is how the main program ends.
}                                    - This signifies the end of the code for main.
```

The code for the program will be typed between the first curly brace ({} and the system("PAUSE") line of code. More complex programs will add other things outside of this area. For now we just want something very basic to run successfully.

For this program there is only one line of code to type.

```
cout << "Hello World!";
```

This is a print statement that prints to the screen. Whatever is between the quotation marks is what will show on the screen when the program runs. There needs to be a semicolon (;) at the end of a line of code, so the compiler knows when to look for the next instruction. Sometimes a line of code might take up more than one line on the screen, so this semicolon helps the computer understand what you are telling it to do.

Once the line of code is typed in, the final program looks like this:

```
#include <iostream>
#include <stdlib.h>
```

```
using namespace std;

int main(int argc, char *argv[])
{
    cout << "Hello World!";
    system("PAUSE");
    return 0;
}
```

The file needs to be saved before it is compiled. Compiling the file turns it into code the computer understands. Click on File – Save and make sure the file is being saved into the project folder created at the beginning of this example. It's okay to leave the name of main.cpp as the file name.

Now we can compile it and run the program. If there are any compiler errors, the program will not run. The most common ones are mistakes in typing or forgetting the semicolon at the end of a line. Click on Execute – Compile and Run. If everything went well, then a console window pops up showing what your program did. My screen output looked like this:

```
Hello World!Press any key to continue . . .
```

The `system("PAUSE");` line of code automatically added the Press any key to continue... part. Notice there is no space between the Hello World! and the automatically added text. This is because we didn't tell the computer to put any space between these two sections of text. Computers only do what the programmers and users tell them to do.

If you want there to be a blank line between Hello World! and the rest of the text, then some code modifications will need to be made. Keep in mind there are multiple ways of doing things in the world of programming. We could add a couple of new line commands (`\n`) in the print text quotation marks or we could add some end line commands (`endl`) outside of the print text.

Let's try using the new line command. Add `\n\n` right after the exclamation point and before the quotation marks. The line should look like this:

```
cout << "Hello World!\n\n";
```

Save the file and recompile and run it to see the changes. This is the output from my program:

Hello World!

Press any key to continue . . .

This is the simplified version of the software cycle. Decide on what to program and which environment to use (write a hello world program using Bloodshed Dev C++). Write the program (type it onto the screen). Compile and Run the program to see if it works. Then make changes to the program based on the outcome of compiling and running it. This can include fixing bugs (mistakes in the program) or making modifications to how the program runs or how the output looks on the screen. This is a continuous process and may include many people and a great deal of time depending on the size of the project.

So far the program only runs in Bloodshed Dev C++. If the program will be run by other people who don't have Bloodshed Dev C++, then it needs to be turned into an executable file. Bloodshed Dev C++ makes this very easy. In fact, this is already done during the compile and run step. Use My Computer to find the project folder you saved at the beginning of this exercise. There will be a picture of a console window for the icon in the folder. If you look at the Details view of the folder contents, it will say it is an application. Double click on the icon to run the console program you just wrote. It looks exactly the same as if you ran it from Bloodshed Dev C++. Now you can let other people run your program from their computers or you can put that executable on your desktop to show people how it works on your computer.

Congratulations on your first C++ program using Bloodshed Dev C++.