

# **Getting Started with Programming in C**

**A series of  
Getting Started Guides  
for  
C/C++ Compilers**

## Contents

Basic Se-tup and Installation of DOSBox.....	3
C Programming with Turbo C++.....	6
Free C/C++ Compliers.....	7
Getting Started with Microsoft Visual C++.....	8
Getting Started with Dev-C++.....	12

# Basic Se-tup and Installation of DOSBox

## Basic Concepts

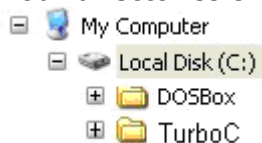
DOSBox is an emulator that recreates a MS-DOS compatible environment (complete with Sound, Input, Graphics and even basic networking). This environment is complete enough to run many MS-DOS programs completely unmodified.

## Installation

The first step is to install DOSBox. This guide will use the 0.72 Windows 32 version which is available for download (<http://www.dosbox.com/wiki/Releases>). Download the release for your operating system. If you are a Windows user, get the Win32 installer. If you are running Vista then look at this site [http://www.dosbox.com/wiki/Dosbox\\_and\\_Vista](http://www.dosbox.com/wiki/Dosbox_and_Vista).

After downloading, install DOSBox to any directory. For example put DOSBox in C:\DOSBox, and Turbo C++ in its own directory from the root. Also make a directory for your programs source code.

Your directories should look something like this:



## Running TurboC

First, run dosbox.exe in your DOSBox folder. In order to make Turbo C available and all your program source code you will set the whole of Drive C as a new directory just for DOSBox. Essentially, it's going to become the C:\> drive of DOSBox. So, type:

```
Z:\>MOUNT C C:\
```

Drive C mounted as local directory C:\

Also by making whole drive available to DOSBox, the paths set for sub-directories within Turbo C are recognised.

## Details of the above command line instruction:

MOUNT      Tells the program to mount a directory

C            Tells the program what you want your new drive to be called (leaving it as C: is fine)

C:\            This is the directory you want to set as the new drive for DOSBox.

After you've done this, you will be prompted with a Z:\>. Now, just change to your new DOSBox drive, which you called C. To navigate to the newly mounted drive just type in:

```
Z:\>C:
```

The prompt should now change to:

```
C:\>
```

To run the Turbo C program you need to change to the appropriate folder / directory, that is you need the TurboC BIN folder in the TurboC folder:

```
C:\>CD turboc\BIN
```

Will now display the prompt as:

```
C:\turboc\BIN>
```

That's it! 'CD' stands for "Change Directory", so you've changed the directory to BIN.

One more step, to run TurboC you need to run the TC batch file, so at the prompt just type:

```
C:\turboc\bin>TC
```

Turbo C will now start and you will be presented with the main IDE screen.

## **Full Screen**

Just press ALT+ENTER to go into and out of full screen.

Full screen can be set to a default by editing the dosbox.conf file. This file is in the DOSBox folder and is a text file that can be opened and saved with any text editor, e.g. Notepad. To default to full screen when you run DOSBox open the dosbox.conf file and change "fullscreen=false" to "fullscreen=true". DOSBox will now run in full screen mode when you open it.

## **Setting a Default application**

If Turbo C is the only Dos program you want to run, you can set up DOSBox to automatically run Turbo C for you when you run DOSBox. In the dosbox.conf, near the end of the file is an 'autoexec' section, edit this section as shown below with the appropriate drive and folder names.

```
[autoexec]
# Lines in this section will be run at startup.
mount c c:
c:
cd turboc\bin
tc
```

Save the dosbox.conf file and run DOSBox, if both the full screen and application changes are made, it will now in full screen and run Turbo C.

# C Programming with Turbo C++

## Getting Started

On this course we recommend that you use Borland's Turbo C++ v3 for DOS as available on the college network. Other compilers can be used and information is available on these but they may not support all the commands taught.

As Turbo C++ is a DOS application you are not able to print from it and may have conflicts with some Windows OS's, e.g. Vista. When using Turbo C++ in the classroom follow the instruction below.

1. Log on the college network with your User ID and Password.
2. Open 'My Computer' or 'Windows Explorer' and select the 'G' drive. Scroll down to the folder called 'turboc3' and copy this folder to your user drive (F) or a USB pen drive (W).
3. Using 'My Computer' or 'Windows Explorer' select the drive to which you have copied the 'turboc3' folder and open it. Now select the 'Bin' folder and double click on the TC file (this is a shortcut file and is only 3K).
4. Set up compiler to use from your USB drive (these details will need to be changed if your Drive has a different designated letter when used at home). From the menu of the TurboC IDE (Integrated Development Environment) select 'Options' and from the drop down list select 'Directories'.

Change the drive letter for the two entries "Include Directories" and "Library Directories".

Add the drive letter for your USB for the "Output Directory" and "Source Directory" entries. Select OK. (You can create sub-folders for your programs reference them here, but they must exist before you add them to the output or source directory path.)

5. Other settings to change are in: Options – Environment – Editor – Default Extension, change this to C from CPP if necessary and in the same window set the tab size to 3 or 4. This will ensure that all your programs are saved as .c files without you specifying the extension.
6. You are now ready to start programming – select File – New and create the template as shown.

Good Luck.....

## Free C/C++ Compilers

This course uses Borland's Turbo C++ for DOS (ver3.1) as its editor and compiler, however, the techniques and structures taught can equally be used with any C/C++ compiler whether for DOS, Windows, Linux or Apple Mac.

Below are links to three FREE C/C++ compilers and a 'Get you started' guide is available for each of them in addition to the Microsoft Visual C++ 6 and Turbo C++ as used on this course.

The MS VC++ 2008 Express and TC++ Explorer version listed below work in the .NET framework and require additional files to be installed. The downloading of these compilers include information on installing the .NET framework.

### **Microsoft Visual C++ 2008 Express**

<http://www.microsoft.com/express/vc/>

Microsoft Visual C++ 2008 Express is currently available for download, free of charge, from Microsoft's site. It allows you to create programs for the .NET framework as well as native Win32 programs.

### **Turbo C++ Explorer**

<http://www.turboexplorer.com/>

Borland's TurboExplorer C++ Explorer Edition allows you to develop C and C++ applications for Windows using the Borland Turbo C / Turbo C++ compiler and IDE. You have to register (free) to get the activation key.

### **Bloodshed Dev-C++ C++ Compiler**

<http://www.bloodshed.net/devcpp.html>

This is a Win32 integrated development environment that includes the egcs C++ compiler and GNU debugger from the Mingw32 environment together with an editor and other facilities to make program development using the Mingw32 gcc compiler easier on a Windows platform. It also includes an installer for your applications.

### **Getting started Guides**

The following pages include basic 'getting started' guides for Microsoft Visual C++ 2008 Express and the Bloodshed Dev-C C++ compiler.

## Getting Started with Microsoft Visual C++

Microsoft Visual C++ is part of the Microsoft Visual Studio suit and the version used in the paper is Version 6. Later versions 'Visual Studio 2008' use the .NET framework and can be downloaded free from the Internet.

To create programs from this course using Microsoft Visual C++ you need to create a 'Console Application', what are 'Console Applications'? In short they are text based programs that run in a DOS type window. There are a number of advantages in creating Console Application, firstly they are simpler, an example is the infamous 'Hello World' in a Console Application could be as short as 5 lines, in a Windows Applications this is likely to run in to a hundred. Even if you use Windows, you will be using some console application, for example many DLL's (dynamic link library) would have been written in C as a console program, or think about the server side script of a web page – in essence they are command line type applications.

The applications that are written with a Windows GUI (Graphical User Interface) could be written in a Console Application would simply would not look as pretty as the graphic capability is not the same – the processing would be exactly the same but in a lot less code.

Win32 Console Applications were often originally written to run under DOS but will run equally well from a Windows console application window. In a Windows NT environment such a program is called a "Win32 console application".

Such software does not have the usual features of Windows software, such as radio buttons and check boxes, but when well-designed it is as easy to use as the usual Windows software — and always takes up a lot less disk space.

Using these console applications makes it easier to learn the language of C and C++ without having to use the Window elements and API. In addition, it allows you to learn the language without being platform dependent.

### Creating a Program

1. Open the MS-Visual C++ application (Figure 1):

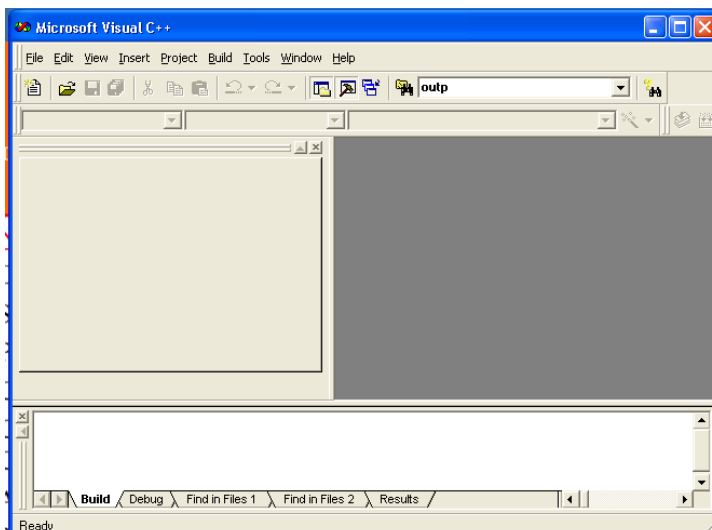


Figure 1

- From the File menu select 'New', the dialog box in figure 2 will open.

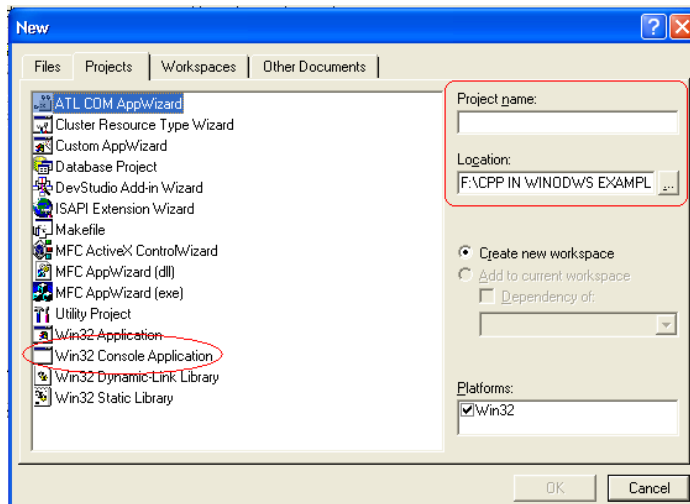


Figure 2

- In this box, firstly select 'Win32 Console Application', enter a project name (we will call this program CPP Demo) and set a location for the project to be stored.

Now click on OK and 'Win32 Console Application – Step 1 of 1' dialog box (figure 3) is displayed.

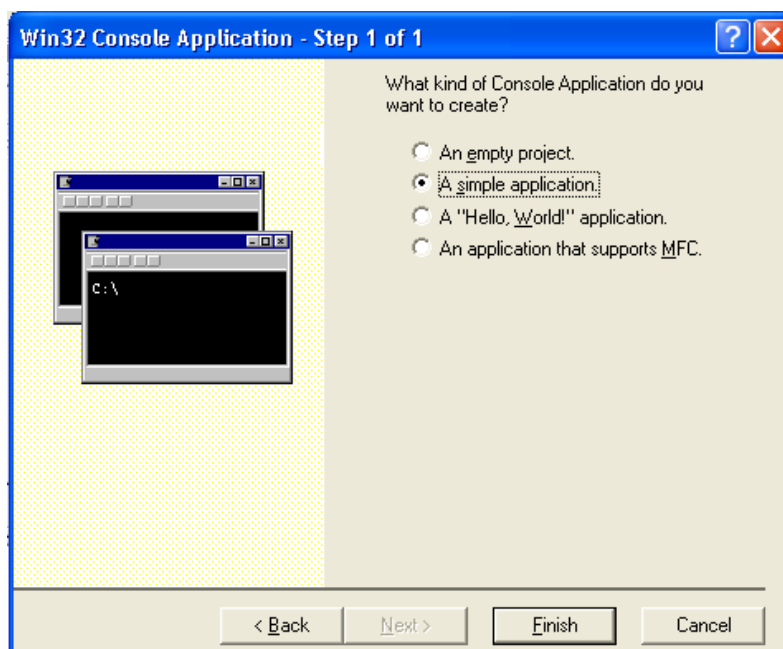


Figure 3

- Click and select the 'A simple application' radio button and then click on 'Finish'.

Then click on 'OK' in the information box that follows. This will now take you to the programming environment.

- You will now be returned to the main application window which is the IDE (Integrated Development Environment) for MSVC++. See figure 4.

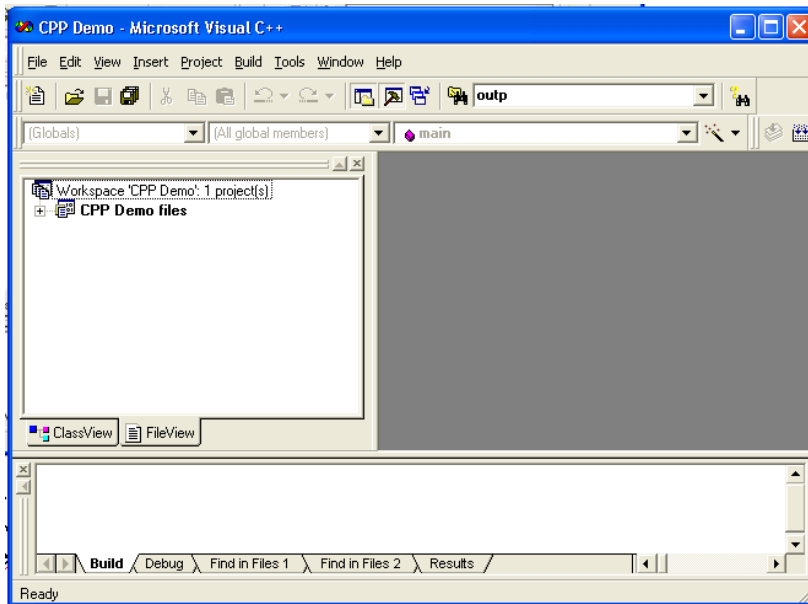


Figure 4

You will notice that there are two file tabs on the bottom of the left-hand window, both will give you access to your source code but it is recommended that you use the FileView tab.

6. Double click on the 'CPP Demo files' item (remember we are using CPP Demo as the project / file name for this example), this will display a full list of folder associated with the project (figure 5).

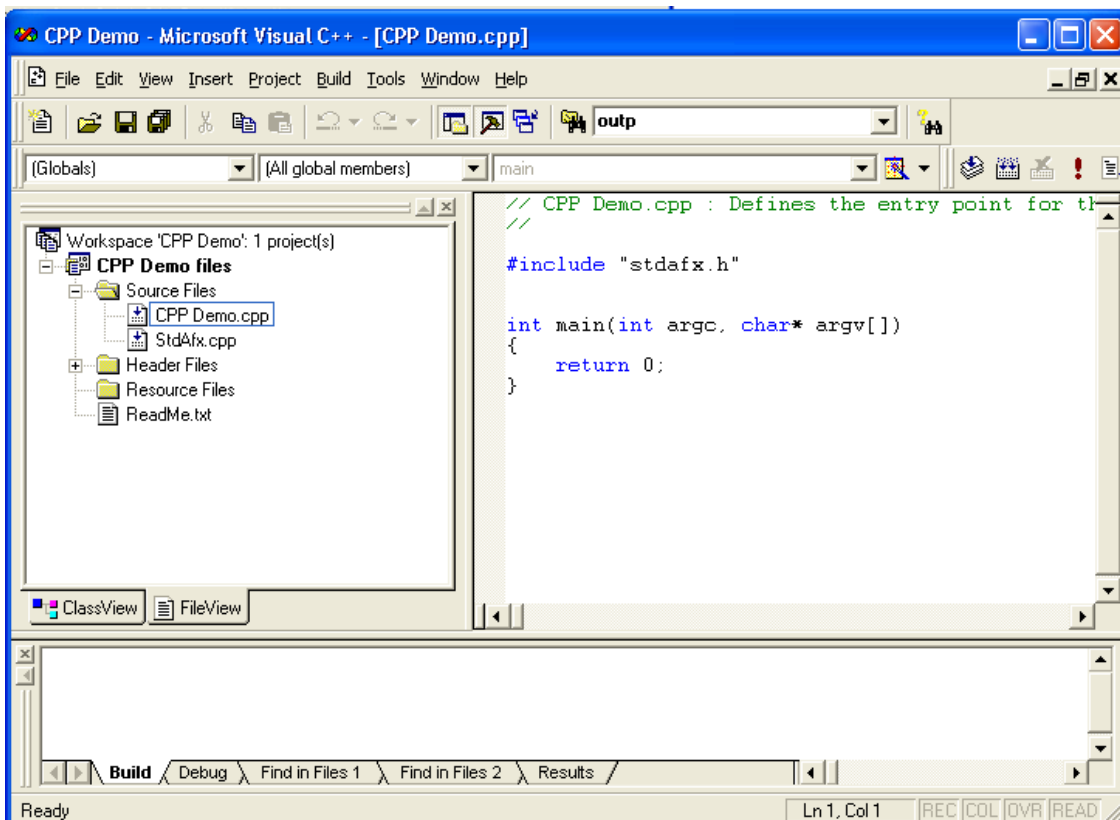


Figure 5

7. To display the skeleton code created by MS-Visual C++, open the 'Source Files' and the appropriate .cpp file (in this case CPP Demo.cpp). Your source code will now be display in the right-hand window (figure 5).

You need to add `#include <stdio.h>` and replace the return data type identifier with `void`. Also remove the parameter of `'main'` and replace with `void`. You can now delete the `'return 0;'`

Try this code:

```
#include "stdafx.h"
#include <stdio.h>

void main (void)
{
    printf("Hello World\n");
}
```

## Getting Started with Dev-C++

Bloodshed Dev-C++ is a free (GPL) full-featured Integrated Development Environment (IDE) for the C/C++ programming language and will run on Windows 95, 98, NT, 2000, XP.

Dev-C++ is available for download at <http://www.bloodshed.net/download.html>

To create programs from this course using DEV-C++ you need to create a 'Console Application', what are 'Console Applications'? In short they are text based programs that run in a DOS type window. There are a number of advantages in creating Console Application, firstly they are simpler, an example is the infamous 'Hello World' in a Console Application could be as short as 5 lines, in a Windows Applications this is likely to run in to a hundred. Even if you use Windows, you will be using some console application, for example many DLL's (dynamic link library) would have been written in C as a console program, or think about the server side script of a web page – in essence they are command line type applications.

The applications that are written with a Windows GUI (Graphical User Interface) could be written in a Console Application would simply would not look as pretty as the graphic capability is not the same – the processing would be exactly the same but in a lot less code.

Now let's get down to writing this Console Application – or at least, get you started. The interface of Dev-C++ is similar to that of MS-VC++ (see handout "Using Microsoft Visual C++").

### Creating a program

- 1 Open Dev-C++ and you will be presented with the IDE – see fig 1.

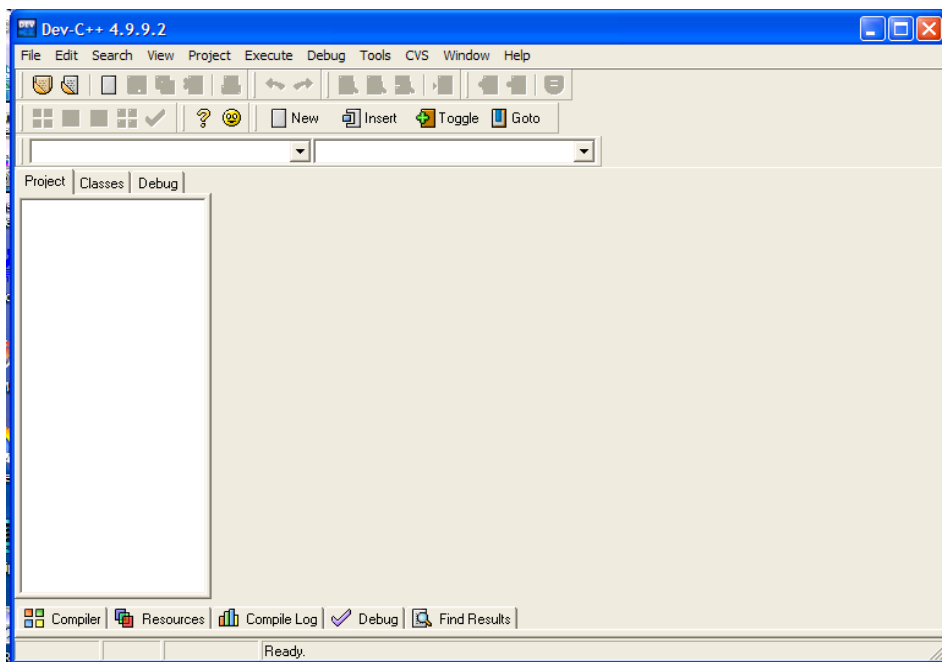


Figure 1

- 2 With this screen open, select 'File' from the menu and from the drop down menu presented select 'New' and then 'Project'. This will cause the dialog box shown in fig 2 to appear. In this dialog box, first select (click on) 'Console Application' icon.

You now need to give your 'Project' a name and select the radio button for a 'C Project', then click 'OK'.

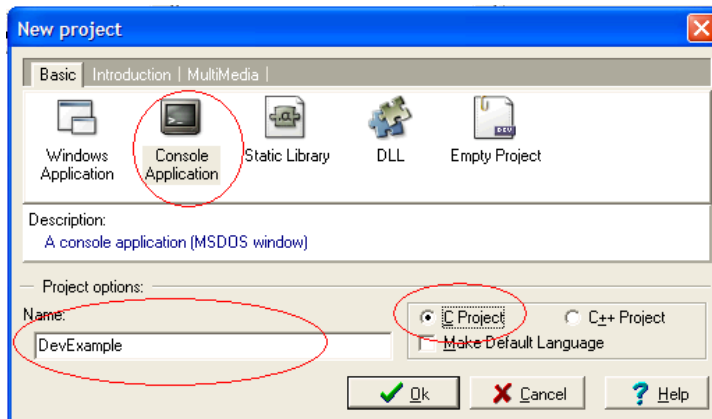


Figure 2

- 3 This will now take you to the editing window (fig 3) and you will find some code already provided. Modify the code to meet your program specification. When your code is complete, before you compile it, SAVE it.

You can now either select the 'Execute' menu and from that select 'Compile & Execute' or the icon which is the 3<sup>rd</sup> from the left on the second row.

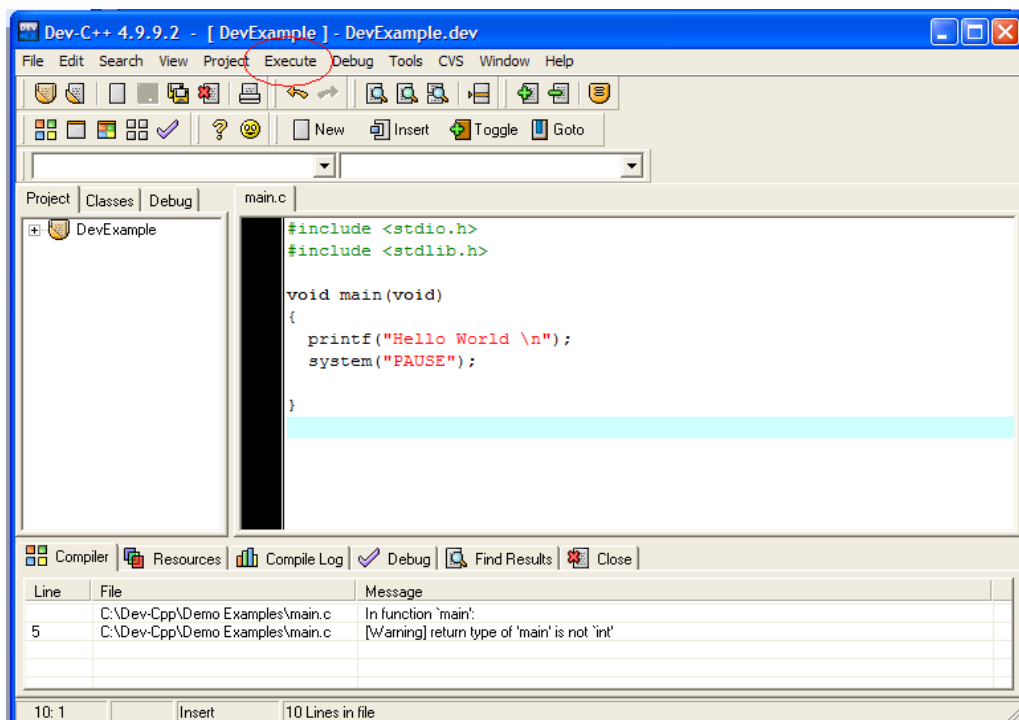


Figure 3