

Introduction

Dev-C++ provides an integrated environment for writing programs. "Integrated environment" means Dev-C++ is a combination program, consisting of a text editor and a C++ compiler. A text editor is a limited word processing program that allows you to type in your program, make corrections and changes, and save and retrieve your program from disk storage. The editor is integrated with the Dev-C++ compiler so that you can easily switch between editing your program and compiling and running it. Note that the Dev-C++ compiler allows you to write, compile and run programs in either C++ or C.

In order to use Dev C++, you must know how to use the computer, how to get around in the integrated environment, and how to use the editor. We've divided this material up into four sections. Everyone should read Sections 1, 2 and 4. The instructions assume you are working in the lab at Brooklyn College; if you are planning to work at home, you should read Section 3 on Downloading and Installing Dev-C++.

- Section 1: The Basics. This is a quick overview of the basics you will need to run your first few programs. This section is a quick summary of what to do.
- Section 2: The Details. This is a more detailed overview of the basics covered in Section 1.
- Section 3: Downloading and Installing Dev C++. This is for those of you who want to work on your computer at home.
- Section 4: Editing, Saving, and Retrieving Files. This discusses editing files and using the File menu.


SECTION 1: THE BASICS

In Section 1, we'll summarize the major steps in using the Dev-C++ integrated environment. Then, in Section 2, we will explain each of these one by one.

1. To enter Dev-C++, find and click on the icon shown at right. It may be in a folder labeled Programming Languages. If there is such a folder, open it by double-clicking on it, and then click the icon shown at right. If you can't find the icon, use the search box and search for Dev-C++.




2. Select File | New | Source File. In the right hand lower window, type in and edit your program using the Dev-C++ editor.

3. Save and name your program by selecting File | Save or by clicking on this icon:  When you name your program, you must specify three things:


- (a) the drive name (nothing, if saving on the hard drive; d: or e: or something else if you have a flash drive)
- (b) the name of the program itself, which should be something like prog1 or first or assn1
- (c) the extension, which must be .cpp for a C++ program.

For example, you might name your first assignment (saved on a flash drive) as follows:
d:prog1.cpp

4. Compile your program by selecting **Execute | Compile** or by clicking on this icon  above the edit window. If your compilation is successful, a Done message will appear; click OK to return to the editor. If your compilation is unsuccessful, an error message will appear; click OK to view a list of errors that you must correct. Use the editor to correct the errors, and repeat step 4 until you are successful.

5. Execute your program by selecting **Execute | Run** or by clicking on this icon:



6. Print your program by selecting **File | Print** or by clicking this icon.  In the window that pops up, remove any check mark in the box that says “Line numbers.” (With line numbers, your printed lines may run off the page.) Click OK.

7. To print your output, place the cursor in the bar at the top of the output window and right-click. From the menu that pops up, place the cursor over **Edit**, and another menu will appear to the right. Move the cursor to the new menu, and click on **Select All**. The window contents will change color. Again place the cursor in the bar at the top of the output window and right-click. From the menu that pops up, select **Edit** and then **Copy**. Then go back to the Dev-C++ environment. Open a new editor window by selecting **File | New | Source file**. Place the cursor in the new window that opens, select **Edit | Paste**, and the output will be copied into the new window. Save it, if you wish, using a name like prog1.out. Then print it as you printed the program file in step 6.

8. Exit from Dev-C++ by selecting **File | Exit**, or by clicking on the Close icon at the top right corner of the Dev-C++ window.



SECTION 2: THE DETAILS



Dev-C++.exe

Step 1: Entering Dev-C++

At Brooklyn College, a shortcut to Dev-C++ has been placed on the desktop. It may be in a folder labeled Programming Languages. Double-click on the folder to open it, find the icon shown above, and double-click on it. In other locations, perhaps at home, to start Dev-C++, you must find it in the Start menu. A typical way to find it is to select Start | All Programs | Dev-C++. You can also search for the program using the Search box, typing in Dev-C++.

After you start the program, the Dev-C++ environment, shown in Figure 1, will appear on your screen. (These screens are from an earlier version of Dev-C++ but are identical except for the version number, which is now 5.3.0.3.) The Dev-C++ main menu selections appear across the top of the screen. Placing the cursor over an icon will cause its use to be displayed at the bottom left of the Dev-C++ window.

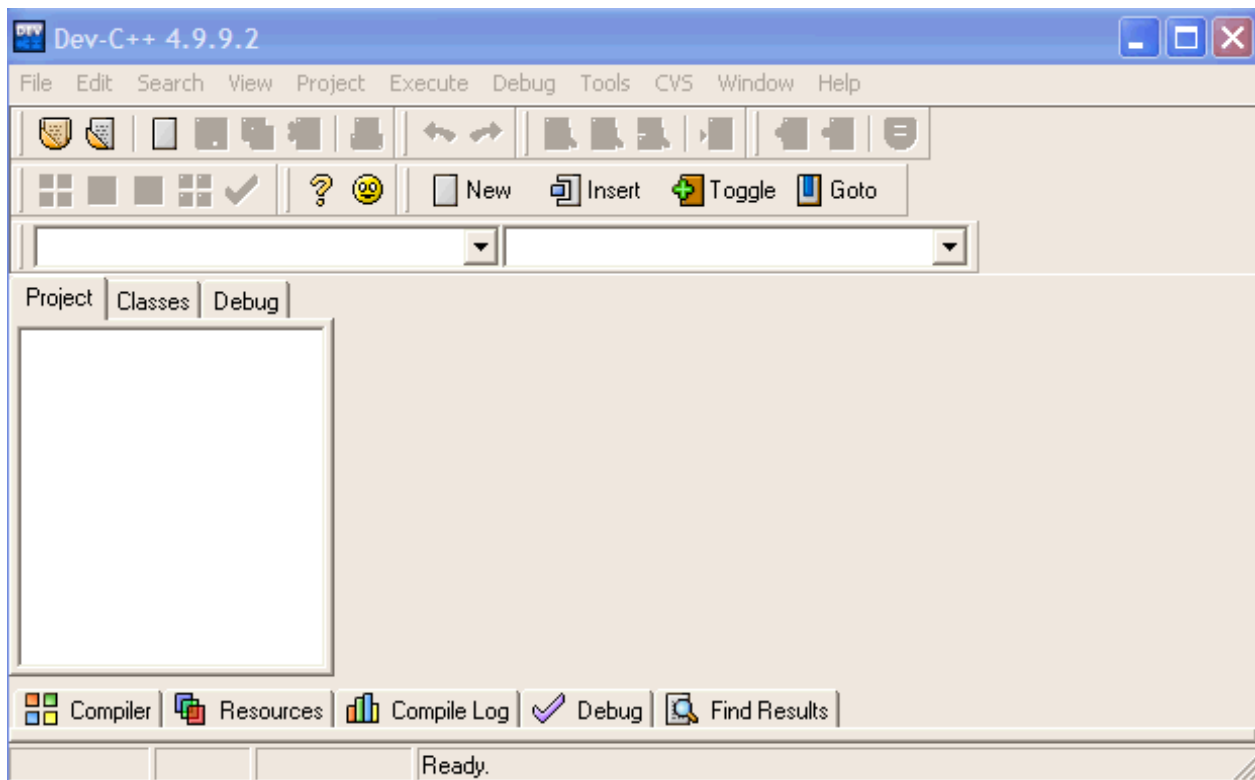


Figure 1 Dev-C++ Environment

For example, placing the cursor on top of the second icon from the left causes the words "Open Project or File" to appear below the icon.



Step 2: Creating a New File

To create a new file, select File from the main menu, then select New from the drop-down menu, and then select Source file from the next drop-down menu (we'll abbreviate this kind of task this way: File | New | Source file).

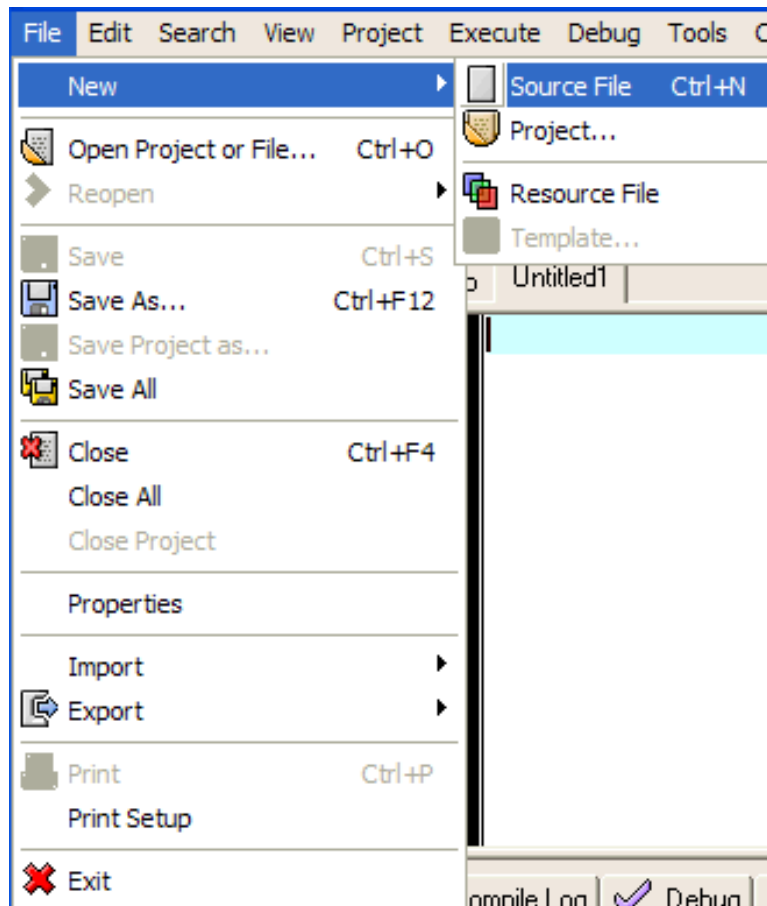


Figure 2 File Menu with Sub-menu

A second window will appear inside the first, and the menu items will change to reflect the options available.

The file you create is initially untitled, as indicated at the top of the editor window in Figure 3.

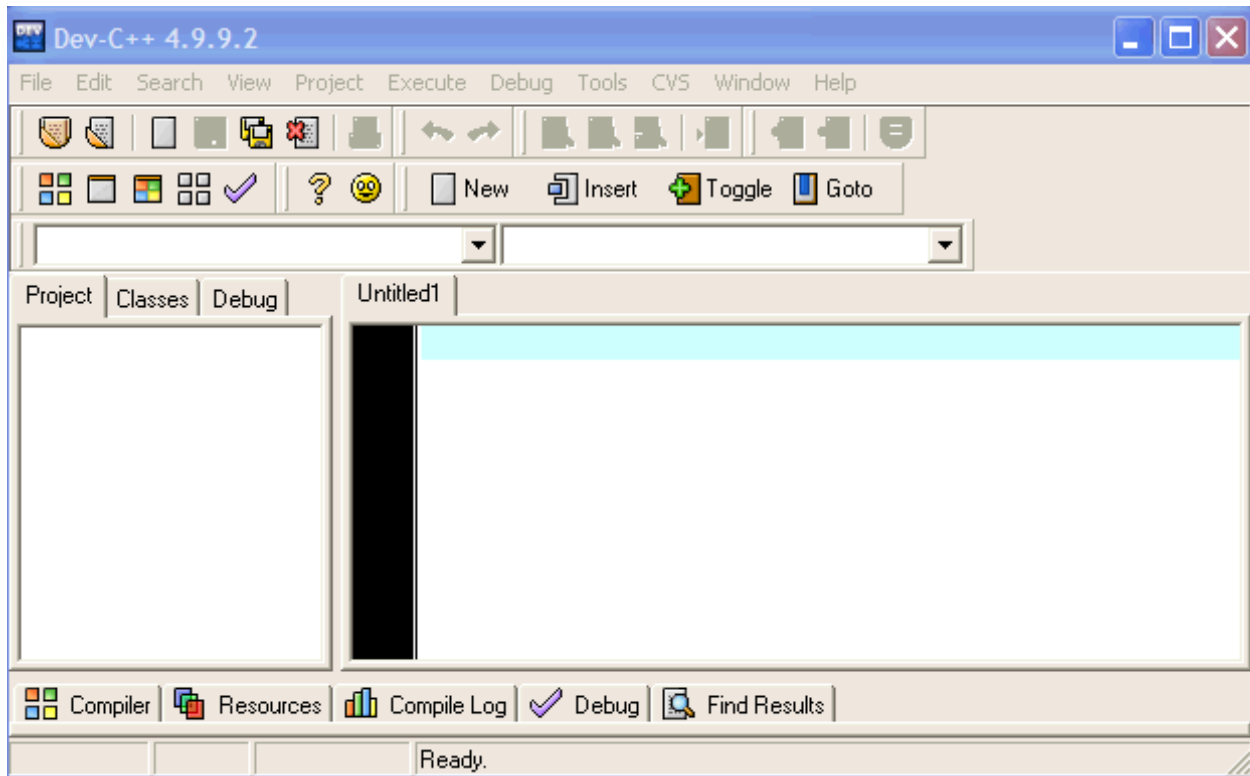


Figure 3 A new file window

As you type in your program, you will notice that Dev-C++ uses different colors for different elements of your program; this is called **syntax highlighting**. The colors can be changed, or you can choose to have everything displayed in one color. Syntax highlighting can help you in debugging your program; for example, if you have typed in a keyword and it doesn't appear in boldface, the word is probably misspelled. If you have forgotten to terminate a comment, all the text following it will be in the same color as the comment.

Step 3: Saving a File



Once you have created a file and typed in a program, you should immediately save it to a disk so that you can use it again. To save a file, click on the Save icon, shown above, or select File | Save. If the file is a new one, the editor will ask you to provide a permanent name for the file before saving it. The "Save File" window shown in Figure 4 will pop up, with the File Name box highlighted. The current filename ("noname00.cpp") will appear in the box. When you start typing, the old name will disappear.

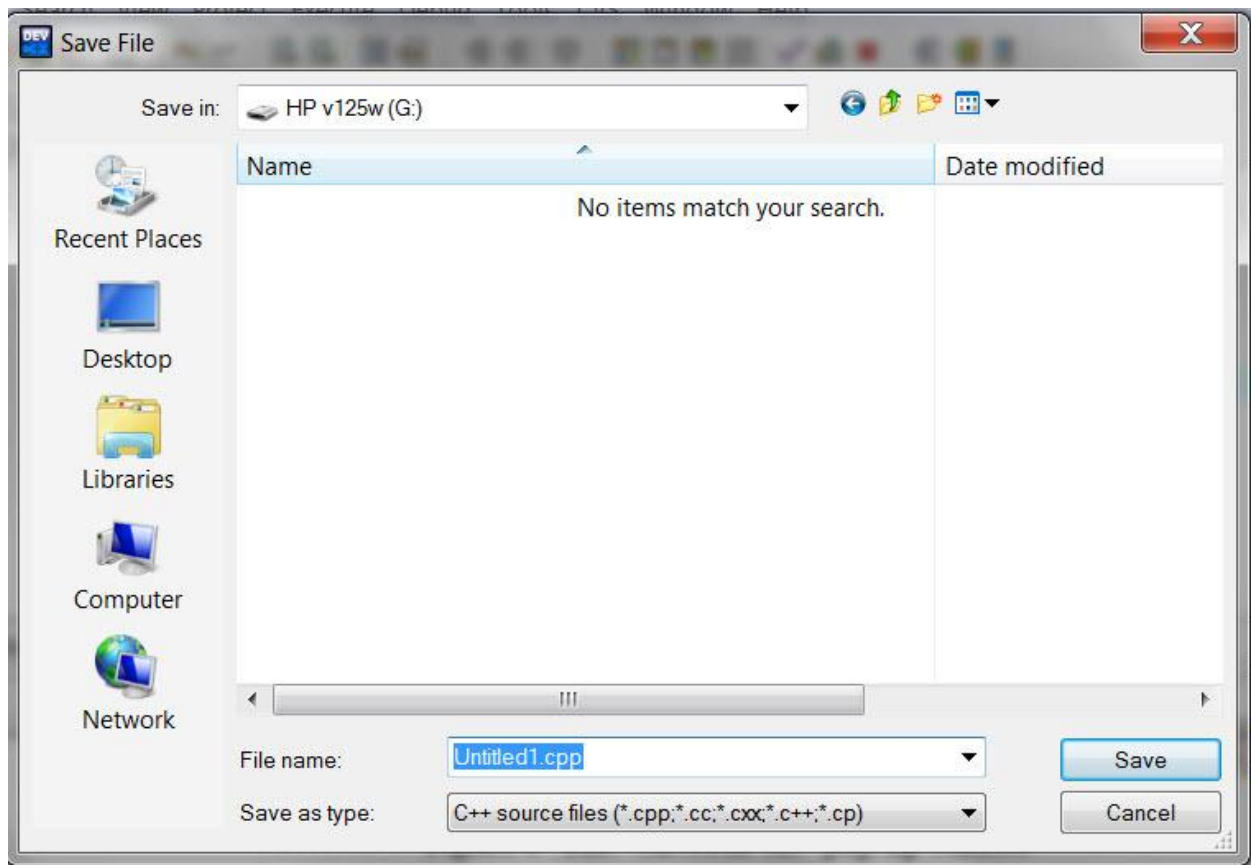


Figure 4 The "Save File As" pop-up window

A file name has three parts: the drive name, the file name, and the file extension. For example, you could use "d:first.cpp" as the name of the file, where "d:" is the drive name assigned to your flash drive, "prog1" is the file name, and ".cpp" is the file extension.

- The small box at the top of the window has a drop-down menu that allows you to choose where to save the file. From this menu, we've chosen the floppy or A: drive. If you are working on a public computer or a network where you should not save to the hard disk, you should supply the name assigned to the flash drive. If you are working at home, it is best to create a specific folder for your work, such as C:\MyPrograms.
- In the first box below the file list, type in a name for your file. The name should reflect what the file is or does. There is no particular limit to the length of the name, but it should consist only of letters, numbers and the underbar symbol; many keyboard symbols are illegal characters in file names.
- In the bottom box, select the type of file you are saving. The selection shown above is the default, and saves your file with the extension .cpp. The extension for a C++ file MUST be .cpp; do not try to change it, or you won't be able to use the program file.

Warning: If you do not specify otherwise, the system will automatically add the file extension ".cpp" to the name of your file. Look at the second item in the file list above; the user tried to name it first.out, but the system automatically added .cpp at the end. If you want to save any type of file other than a program file (such as an output file or a data file), select "All files (*.*)" from the bottom box's drop-down menu. Then you can specify your own preferred extension, like .dat for a data file or .out for an output file.

To leave the "Save File" window and save your file, press <Enter> or click **Save** with the mouse. You can leave the window without performing any action by selecting **Cancel**.

Once you have named a file, subsequent Save operations will not bring up the File Save window. Changes will be noted only by two small changes in the appearance of the Dev-C++ window. The Save icon will become gray to indicate that the save action can not be performed at this time. In addition, the word "Modified" will disappear from the status line at the bottom of the window.

WE RECOMMEND THAT YOU SAVE YOUR FILE EVERY FIVE OR TEN MINUTES.



If you do not save your file, and the computer crashes or someone kicks out the plug, you will lose all your unsaved work. It is better to lose less work by saving every few minutes.

Dev-C++ automatically saves your file before compiling it. This can cause two surprises: 1) the Save File window may appear asking you to name the file, and 2) if you are making some "what if I do this" kinds of changes, you may lose the original version of your program. If you want to keep more than one version of a program, use the File | Save as feature to save the file with a different name.

Step 4: Compiling a Program



After you have written a Dev-C++ program, you will want to compile and then run it. To compile your program, first save it, and then click on the Compile icon, shown above, or select Execute | Compile. This invokes the Dev-C++ compiler, which will compile (translate) your program into one which the computer can understand. If the compilation is successful, the box shown in Figure 5 will appear. Close it and run your program.

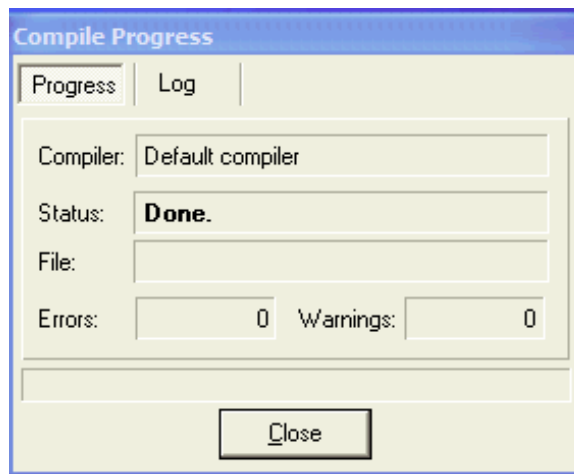


Figure 5 Successful Compilation Status Window

However, if the compiler finds any errors during the compilation of your program, it will not display this box, but will instead provide a list of error messages and highlight the first line with an error, as shown in Figure 6.

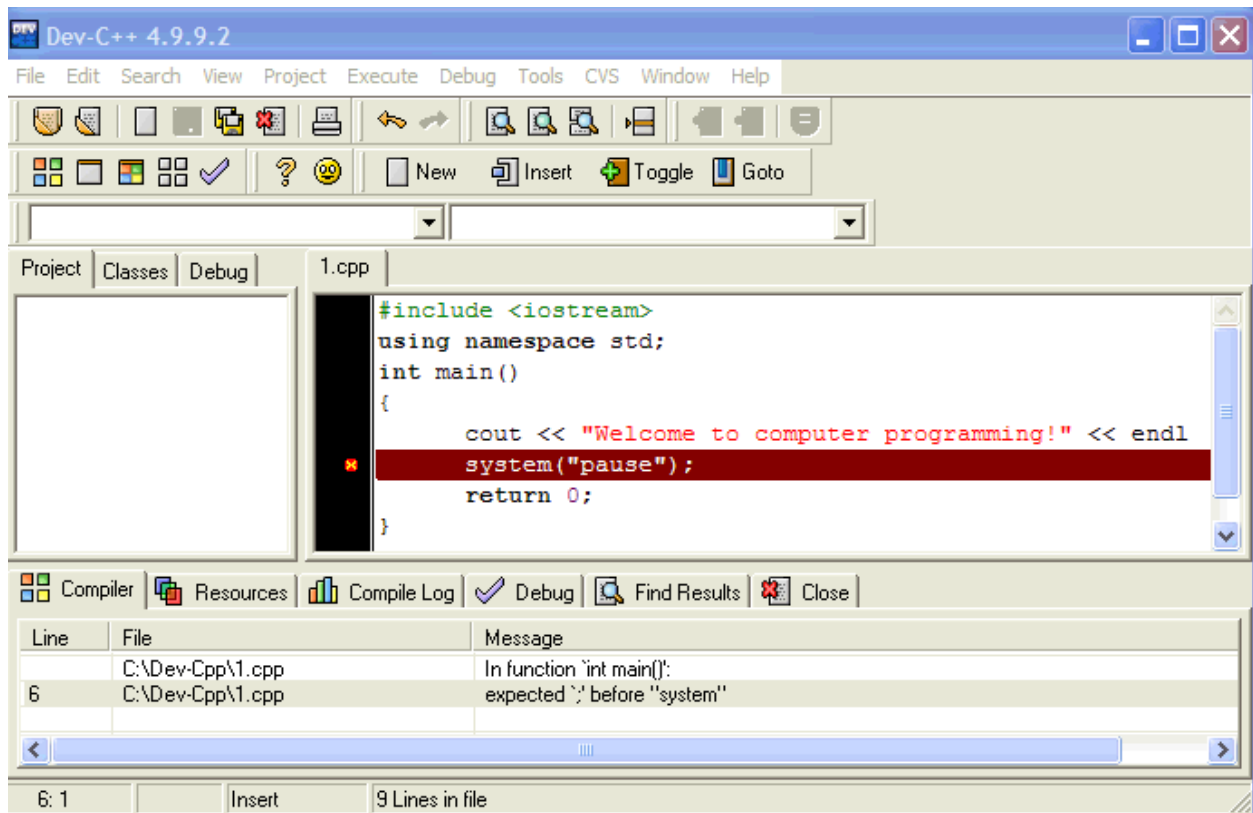


Figure 6 A window showing compilation errors

The bottom window contains a list of all your errors. You can scroll through the Message window to see all of the errors. Clicking on an error message highlights the relevant line in the top window.

Once you have located the error, fix it. Then return to the Message window to view the next error. Continue the process until you have corrected all the errors. Then again save and compile your program. Repeat this process until your program is free of errors.

If you cannot identify the error from the message given by the compiler, or if you do not know how to correct the error, take a printed copy of your program and the error message to your instructor. (Error messages are not always perfectly clear, because they are based on trying to translate something that doesn't make sense.)

If you have corrected all the compilation errors in your program, it will compile successfully, and the window shown in Figure 5 will appear on the screen.

Step 5: Running a Program



To run your program after compilation, click on the Run icon, shown above, or select Execute | Run. If your program is correct---and if you remembered to include the line

```
system("pause");
```

right before return 0; --a window will appear in which you can review the results, or output, from running your program. Look at the output from running your program, to see that the program did what you expected it to do. If it did not, correct any mistakes that you can identify, then recompile and rerun the program. (You must close the output window before you can run your program again.)

Errors may be discovered as your program is running; these are called **execution errors**. Some execution errors may be identified by error messages in the Message window, as in Figure 6. Other errors may show up by an absence of output, or by output which is not correct. Some errors show up because the computer never stops executing the program (this is called an **infinite loop**). Once again, if you cannot identify the errors, print the program and the output and seek help.

If the program ran correctly and gave the correct output, make a copy of the program and the output to turn in to your instructor for grading. **Do not erase your file from your disk.** If your instructor finds any errors, you may have to modify and rerun the program. In addition, you should save all graded programs until the end of the semester in case your grade is lost.

What to Do About an Infinite Loop, or Worse

Your program might have an infinite loop or, worse, might cause the machine to **freeze** or **hang**, which means it will stop responding to your commands. Here are some suggestions of what to do if something like this happens.

- (1) Breaking out of an infinite loop which prints is easy: simply close the output window, and the program will stop running.
- (2) If that doesn't work, try to break out of the program by pressing **<Ctrl>-<Break>**. If that doesn't work, try **<Ctrl>-C**. You may have to do this several times.
- (3) If that doesn't work, try to end the task by pressing the **<Ctrl>**, **<Alt>** and **<Delete>** keys together. The Task Manager (in Windows XP) will open, and you can use the mouse to highlight the name of the offending program, then select "End task" to terminate it.

Compile and Run



To save time, it is possible to compile and run your program in one step. To do this, select Execute | Compile and Run or select the icon shown above. If the program compiles successfully, the computer will go ahead and run it. Of course, if the program doesn't compile, the process will stop and generate error messages. At least at the beginning, it is preferable to compile and run in two separate steps.

Step 6: Printing Your Program



To print your program, click on the icon shown above or select File | Print. First, save the program, because occasionally printer errors can cause you to lose the active program. Make sure that the printer is turned on and that your program is in the active window (its tab is forward). Then select File | Print. Select options if you wish, and then click OK. Your program will print on the printer.

Step 7: Printing the Output from Your Program

Unless you specify otherwise, the output from your program will appear in a window on the screen. Until your program is working perfectly, this is fine. But once the program is working correctly, you will want to print the output.

If your output window is full-screen (that is, it fills your computer screen), first reduce it to window-size by holding down the <Alt> key and pressing the <Enter> key. Then follow the steps below.

Right-click in the colored bar at the top of the output window. The menu shown in Figure 7 will appear.

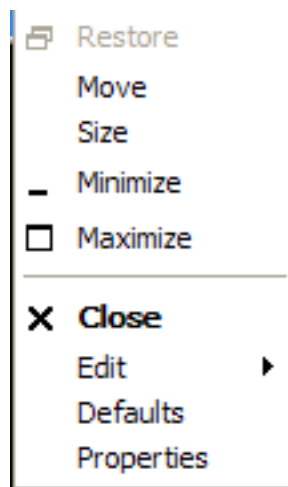


Figure 7 Menu from right-clicking output window

From this menu, move the cursor to Edit, which will bring up the sub-menu shown in Figure 8.

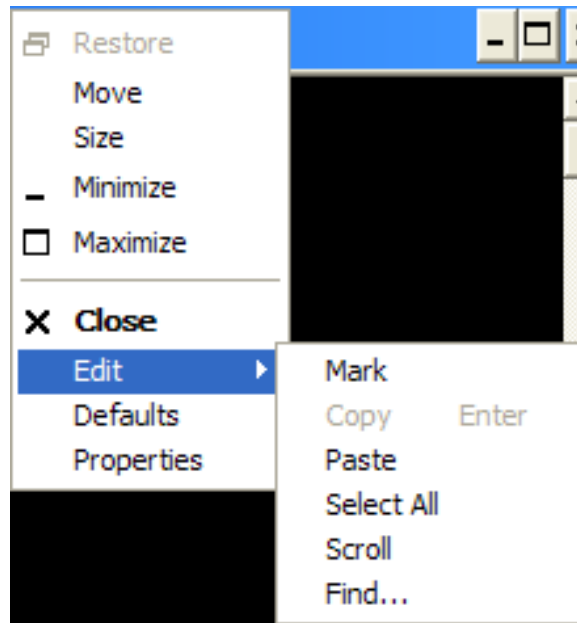


Figure 8 Edit menu from output window

From this menu, left click on Select All. The contents of the output window contents will change to black on white. Again place the cursor in the bar at the top of the output window and right-click to get the menu shown in Figure 7. Move the cursor to Edit, and from the menu shown in Figure 8, select Copy.

Then go back to the Dev-C++ environment. Open a new editor window by selecting File | New | Source file. Place the cursor in the new window that opens, select Edit | Paste, and the output will be copied into the new window. Save it, if you wish, using a name like a:prog1.out. Then print it as you printed the program file in step 7.

Since the output produced by this method is sometimes not very attractive, your instructor may show you a better method.

Step 8: Quitting Dev-C++

Of course, at some point, you will want to quit Dev-C++. Select File | Exit, or click on the Close Program icon shown above, which appears in the upper right-hand corner of the Dev-C++ window. If you have an open and unsaved file, Dev-C++ will check to see if you want to save the file before quitting (see Figure 9):

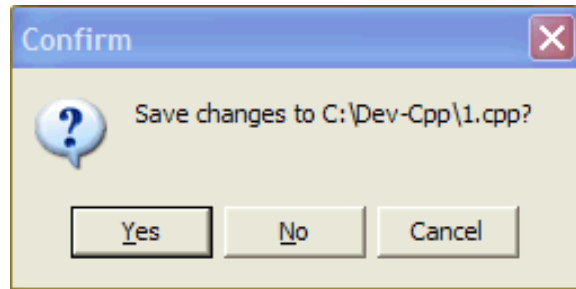


Figure 9 Save before closing dialogue box

Choose Yes or No as appropriate, or Cancel if you didn't intend to close Dev-C++.

Sample Program

Here is a complete sample C++ program. You should enter, compile and run it. This program sends one line of output to the screen.

```
/* sample C++ program */
#include <iostream>
using namespace std;
int main()
{
    cout << "my first C++ program" << endl;
    system("pause");
    return 0;
}
```

SECTION 3: DOWNLOADING AND INSTALLING DEV-C++

One of the advantages of Dev-C++ is that it is free to download and use. If you want to work at home, you should download the program and install it on your home computer.

Web Site

The compiler can be downloaded from this location:

<http://sourceforge.net/projects/orwelldevcpp>

Downloading the File

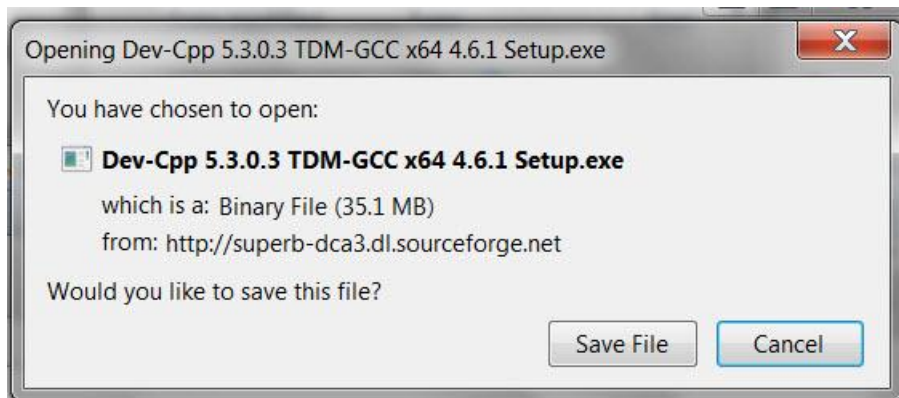
The project page looks like this.



Click on the word "Download" in the green box. A new page will appear with these words: "Your download should begin shortly." Typically, your download will begin automatically.

Downloading Using Firefox

In Firefox, this window will appear:



Choose Save File.

A Downloads window will appear with a green progress bar.



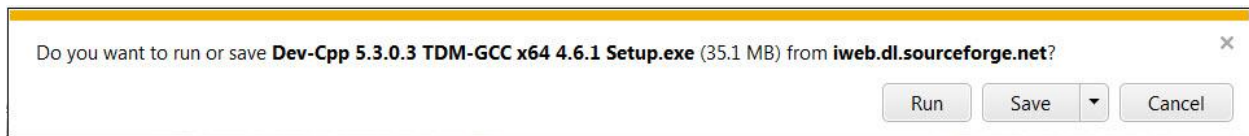
When the download has finished, the name of the setup file will appear in place of the progress bar.



Click on the name of the setup file to start the installation.

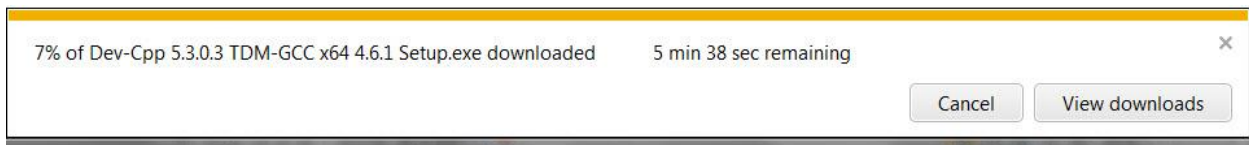
Downloading Using Internet Explorer

In Internet Explorer, you will get this box at the bottom of the Explorer window.

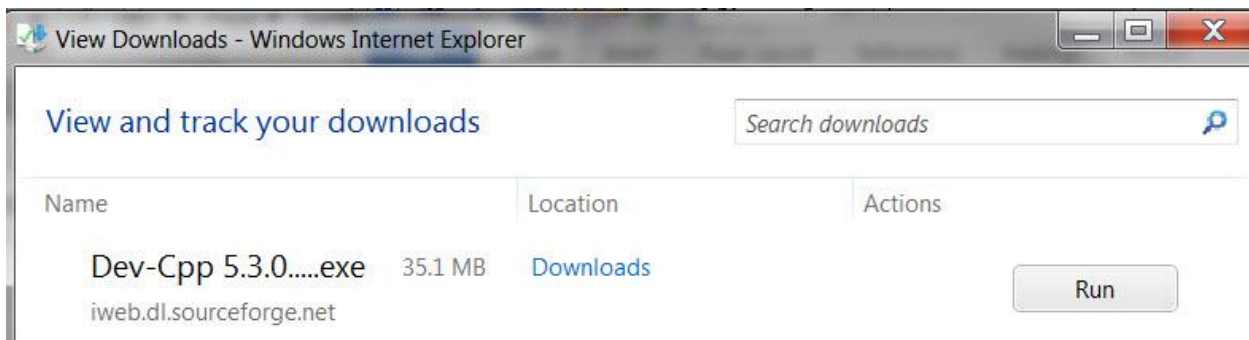


You can choose Save or Run. Choosing Save stores the downloaded file, while choosing Run does not. We've chosen Save. If you choose Save, you will have to run the setup separately after the download.

When the download starts, you can watch progress by percentage at the bottom of the window:



It saves the file in your chosen download folder (usually Downloads) When done, you will see this window.



Click Run to start installation.

Installation and Configuration

Follow the prompts to allow the program to make changes to the computer. The setup will unpack its files. You'll be asked to choose a language; choose what you prefer. Click "I agree" to accept the license agreement. When asked to choose components, leave the choices as is and click Next.

A Choose Install Location will appear, offering to install the program in a default folder. Accept this by clicking Install (or use Browse to select a new folder and then click on Install).

After the files are installed, a window will appear telling you that the Setup Wizard has completed your installation of Dev-C++. There is a box checked that will open Dev-C++ for the first time. Leave it checked and click Finish.

Dev-C++ will start. Several windows will appear asking you setup information. Click Next on all of them. Keep the New Look interface and the Classic Plus Editor and click Next. Everything you select can be changed later in Preferences if you are unhappy with your settings. Click OK at the end.

Done!

A window will appear saying that Dev-C++ has been configured successfully. Click OK, and the program will start.

A "Tip of the Day" window will appear. Click in the box next to "Don't display tips at startup" to make this stop, and click Close.

Now you are at the window shown in Section 1, Figure 1.

SECTION 4: EDITING, SAVING, AND RETRIEVING FILES

Files and the File Menu

All actions pertaining to creating, saving, naming and retrieving files are performed through the File menu, shown in Figure 10.

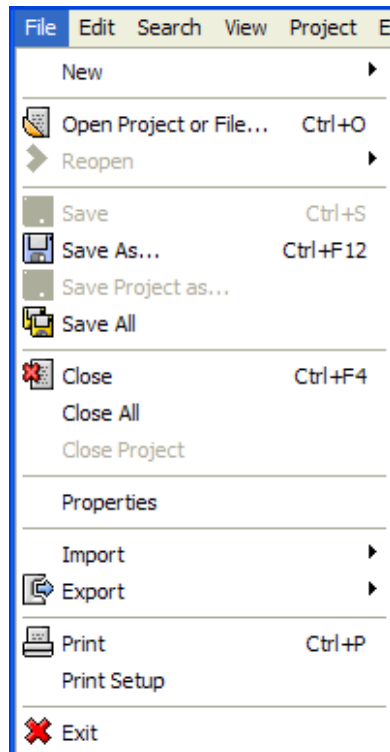


Figure 10 File Menu

Selecting and Opening an Existing File

Once you have written a few programs, you may wish to work on one of your existing files instead of creating a new one. To do this, select File | Open Project or File. This pops up a window which allows you to choose an existing file (see Figure 11).

In the top box, you will need to specify the folder or drive in which to find the file. You can do this by means of the drop-down menu or by searching through the folder list in the main window. You will need to specify or find the name of the file (in this case 1.cpp), and specify what types of files to examine. Typically, saving all your files in one folder or on a flash drive will make this task quite simple.

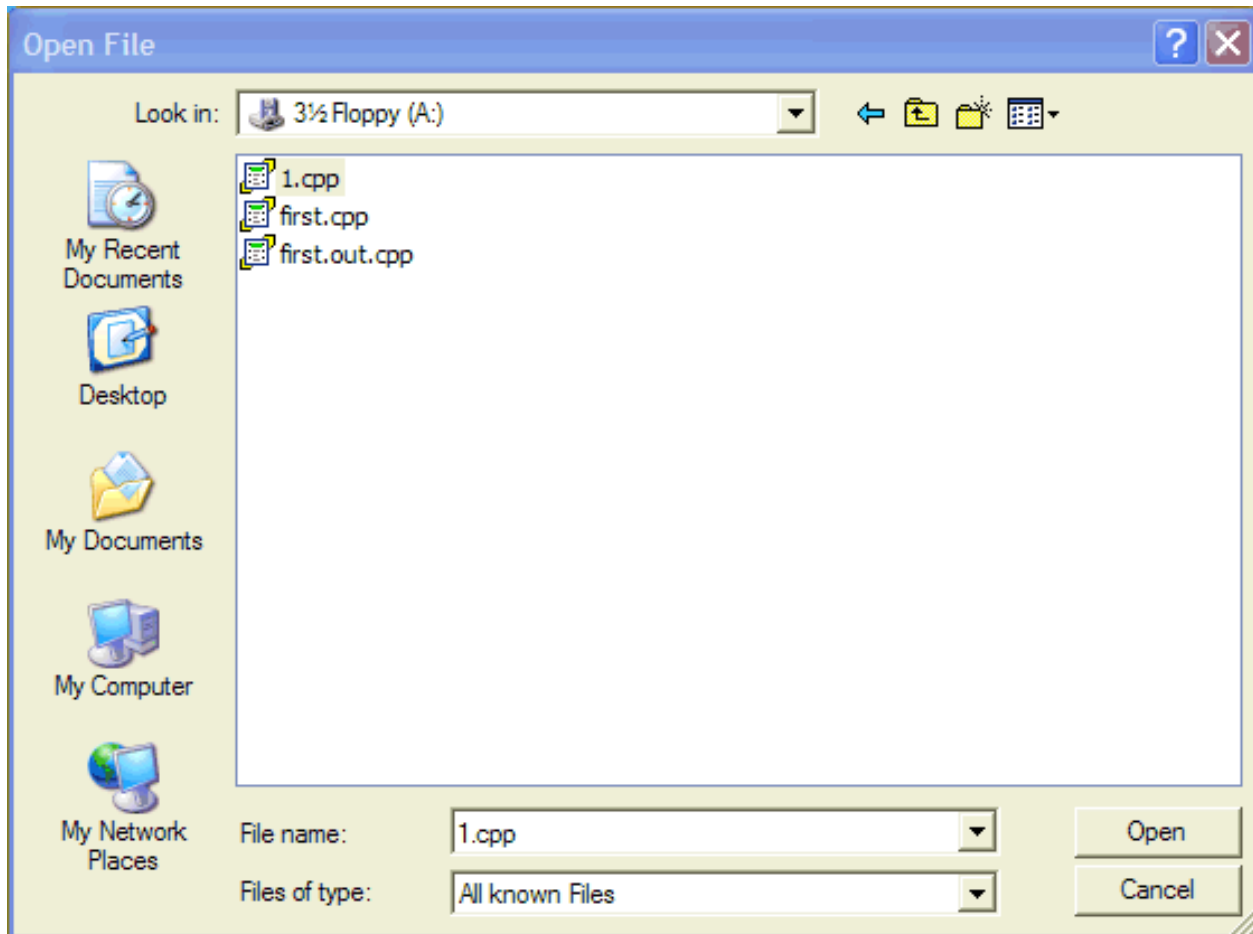


Figure 11 File Selection Window

Changing the Folder

You can change the directory to the drive and directory where your files are stored. To do this, move to the Look in: input box, click on the arrow to the right of the box to bring down a selection of drive names, and select the drive on which your files are stored. Generally, in a computer lab, this will be drive a: or the name associated with your flash drive. Click on the drive name to select it. Once you have done so, the entries in the main window above will change to show the folders and files on the selected drive. If the drive has any folders, select the appropriate folder from the list by double-clicking on it. A list of files will appear in the main window. They will all have the type indicated in the Files of type input box; if you are looking for a C++ file, you should specify "All known files," or "C++ source files" (these will be files with extensions like ".c" and ".cpp").

To get a list of all your files (with any extension), select All files (*.*) in the File of type input box. Then select the file you wish from the main window.

Saving a File under a New Name

Sometimes, you may wish to keep two distinct versions of a file (perhaps the new version

tests some changes that you aren't quite sure of). To do this, use the Save as selection from the File menu. Save as allows you to save a file under a name other than its original name. The dialog box that appears will be identical to the one shown in Figure 4, and the actions you take will be similar. Later, you can retrieve either file.

Creating a Data File: Use Notepad

Many of the later programs require you to create a data file. This file, which should usually be stored on the same disk as the program itself, will hold the data that the program will read.

You can create a data file in the Dev-C++ editor (using File | New | Source file). However, we do NOT recommend this. When you save the file, the Dev-C++ editor inserts two characters to mark the end of the file. This can cause problems when trying to read to the end of a file as you will do in later programs.

Instead, we recommend using Notepad to create a data file. To do so, find and open Notepad. (you can find it by clicking Start | All Program | Accessories).

To create the data file, select File | New. Type in the data as directed by the assignment. Then save the file, giving it a name which will help you remember what program it goes with. For example, if you are writing a program called "a:program7.cpp", you might want to call its data file "a:prog7.dat".

If you do not specify otherwise, Notepad will automatically save the file with the extension ".txt". Choose the folder in which you'd like to save the file (it should be the same location as your program file) and click Save.

If you want to save a file with any other extension (such as .dat), select "All files (*.*)" from the "Save as type" drop-down box. Then you can specify your own preferred extension, like .dat for a data file. To open such a file later on, you might be asked to choose a program with which to open the file; if this happens, choose Notepad and check "Always use this program to open this type of file."

If you do create a data file using the Dev-C++ editor, you should know that if you do not specify otherwise, the editor will automatically add the file extension ".cpp" to the name of your file. If you want to save any other type of file (such as a data file), select "All files (*.*)" from the Save menu's bottom box's drop-down menu (see Figure 3). Then you can specify your own preferred extension, like .dat for a data file.

To print a data file, in either Notepad or Dev-C++, open the file and select File | Print.