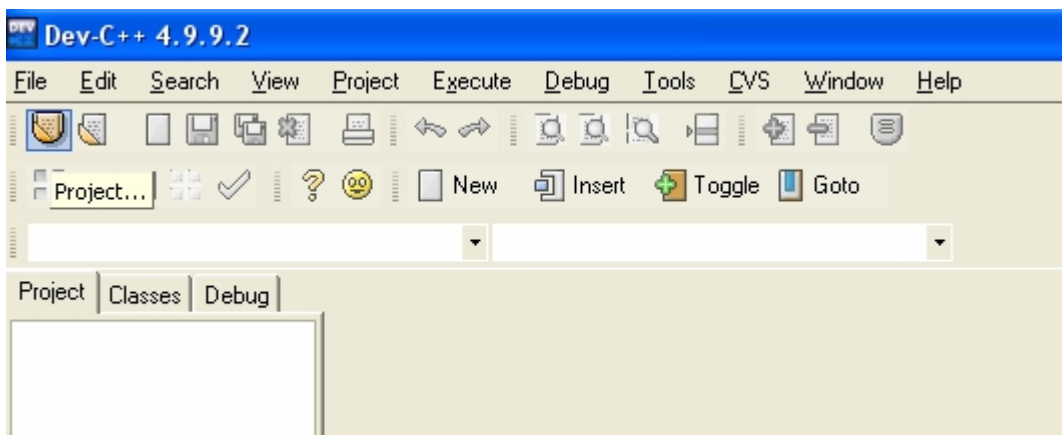


A Quick Tour of DevC++

This short tour gives new users a glimpse of the DevC++ Integrated Development Environment (IDE). The IDE is the center of DevC++ software development. From the IDE you edit, browse, compile, link, debug, and do many other programming tasks. Before you can begin learning about the DevC++ IDE, you'll need to launch it.

Click the **Start** button in the task bar, choose **Programs**, choose **DevC++**, Alternately, double-click on the desktop icon for **DevC++**. The result is the window shown below. Positioning the cursor over active menu buttons pops up a small flag indicating its use. The icon on the left starts a New Project for instance.



After starting the DevC++ program in the usual Windows XP manner you will find the IDE is made up of these parts:

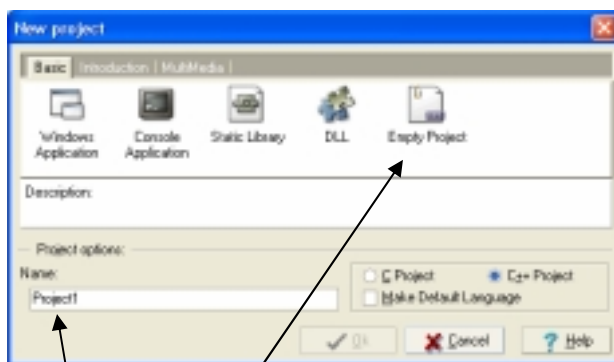
- **An editor** – So you can write and modify your program without leaving the environment
- **A compiler** – So you can compile your program (or find syntax errors that prevent your program from compiling)
- **Resource Workshop** – Lets you create and edit the resources (dialog boxes and menus, for example) that your programs need
- **An integrated debugger** – So you can find and correct mistakes
- **A project manager** – So you can easily build executables (programs) and libraries
- **A browser** – So you can understand the relationships of the various objects in object-oriented programs
- **Options notebooks** – So you can easily control the behavior of all aspects of the IDE
- **A scripting language** – to let you customize how the IDE works
- **An integrated Help system** – So you can get more information about using the IDE or creating C++ programs without needing to open a manual

You will use the menus and SpeedBar buttons to access these features – causing various new windows to appear in the IDE. For example, when you use the editor, an edit window appears. And when you use the compiler, a dialog box appears, showing the status of the compiler. Flyby Help hints appear on the status bar (the gray bar that stretches across the bottom of the IDE). If you move the mouse across menu commands, the flyby hints explain what the menu items do. If you are in the middle of opening a project, the hints help you with the various steps.

What we will be doing

Creating windows programs is beyond the scope of this course. Many things are required just to get a simple window to appear on the screen. DevC++ provides lots of features to assist in this task, but in this course we will be primarily concerned with developing console programs of relatively few lines of code. Most commercial Windows applications often require 30,000 –70,000 lines of code! DevC++ lets you describe the basic features of the application you want to create, and then creates the foundation lines of code for you, *but we are a long way from being able to do this*. Some programs consist of a single file, but most programs are larger and involve many different source-code files and many different header files and libraries. To create the final executable, you need to compile each of these different source-code files and then link them. Fortunately, project files and the IDE provide the capability to do this without requiring you to know all the details about how the compiler goes about compiling and linking a program.

Creating a New Project

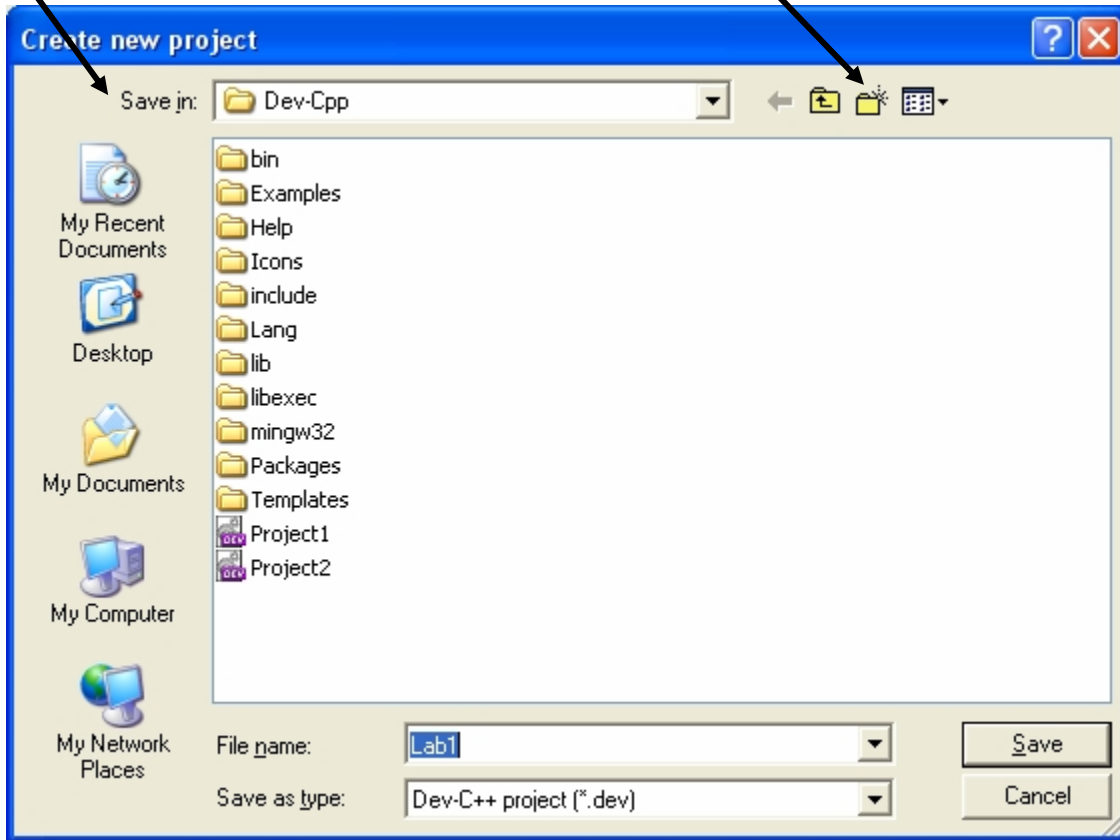


Select Empty Project

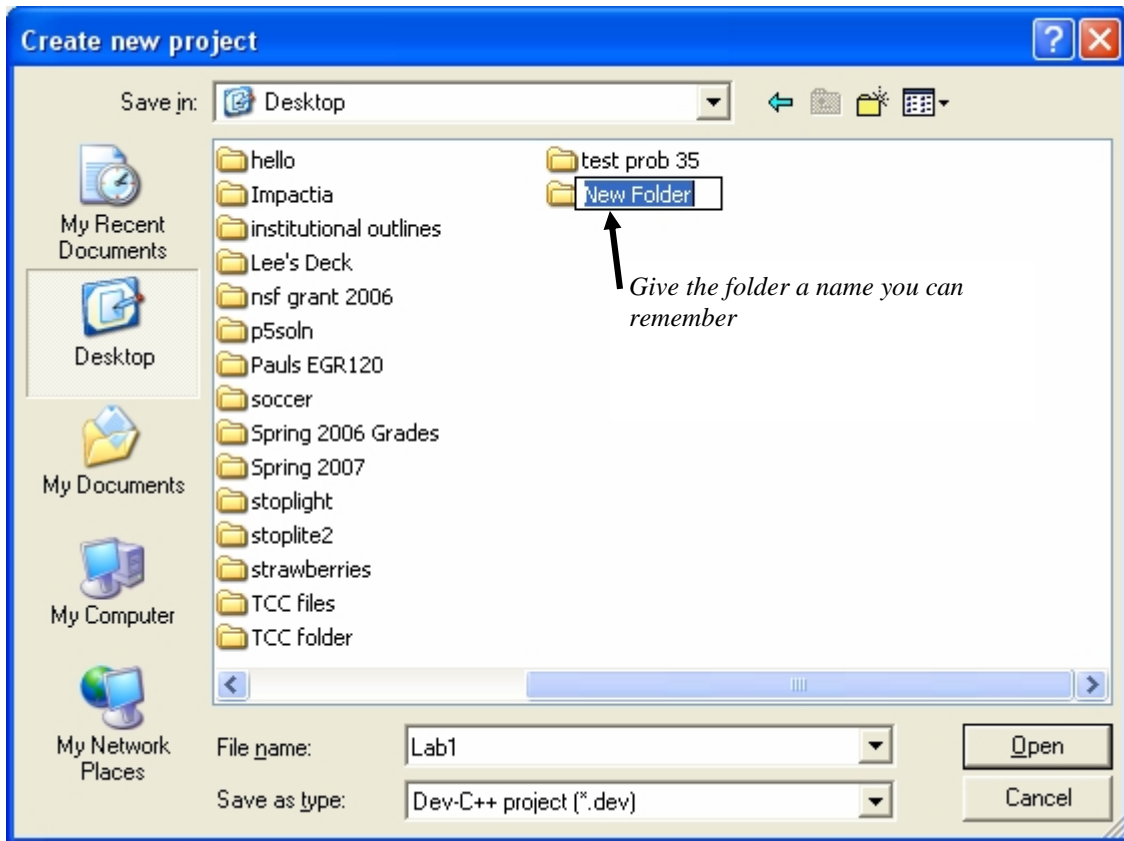
and then give it a name, like Lab1 or LoopHW

You need to create a project file whenever you build a program. The project file tells the compiler what source files to compile when building an application. The project file also tells the compiler what libraries to link in. Creating a new project is easy. Just select **File** → **New** to bring up the New Project dialog box. We will be targeting to applications that run on PC's. Next we need to give the project a name. You will be asked to Name the Project.

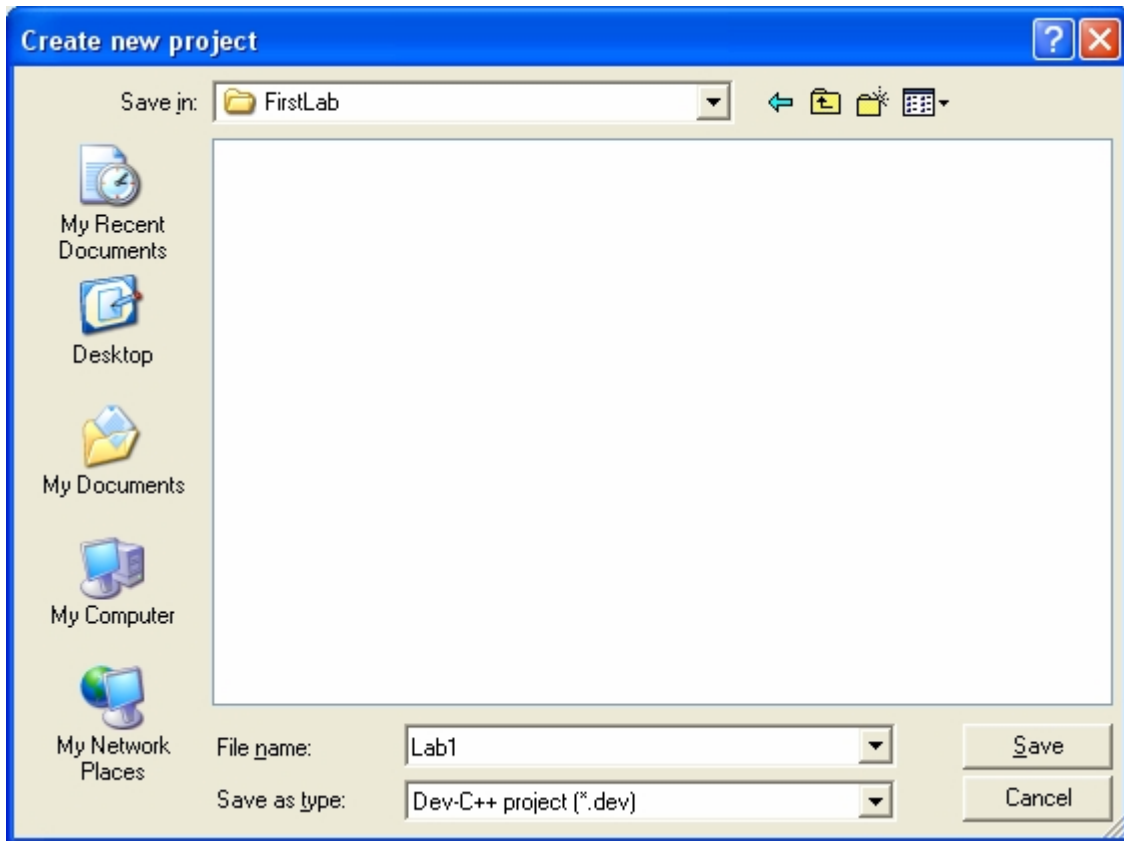
Next you will pick a location to save the new project. Change the Save in window selection to a location of your choice. In the H-151 lab use the desktop to place your folder so you won't lose it (you can save it to your media later by dragging the folder to your jump drive, CD, etc.) Next, create a new folder.



Here, under "File name:", type in the name of our first lab exercise project name, Lab1. Note that the path for our project directs it to be saved on the computer's desktop.

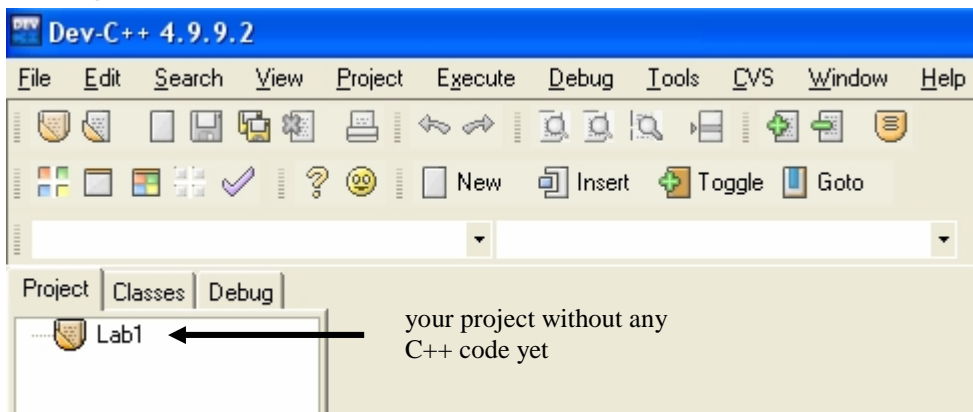


You should always save your work on your own media in the TCC computer lab because you probably will not find your work again on the hard drive the next time you use the lab. The H-151 lab erases student files automatically. It also lets you work on your project in other labs or at home if you have DevC++ installed there.



Clicking the Save button now will put your new Lab1.dev project file in a folder on your desktop called FirstLab.

Using the Project Window

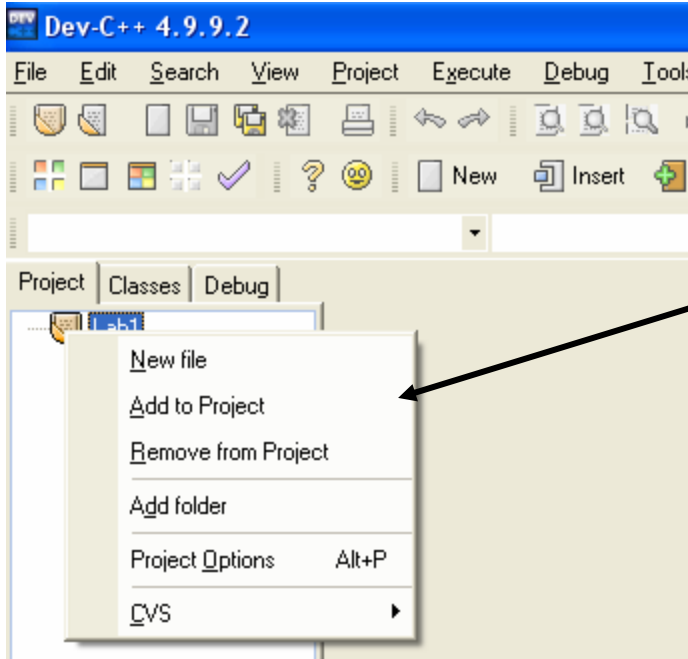


To effectively use DevC++ IDE, you should be familiar with the Project Window. The window displays lots of information about a project, so identifying and understanding that information is our next step. The project window has these items:

Project Classes Debug

Project Tab

The project window uses view tabs to present different facets of information in the project. Right Click a project tab to change the information displayed in the project window.



Right clicking on the Project tab brings up many options. You will frequently want to use the New file option to add another source file to your projects.

Classes and Debug Tabs

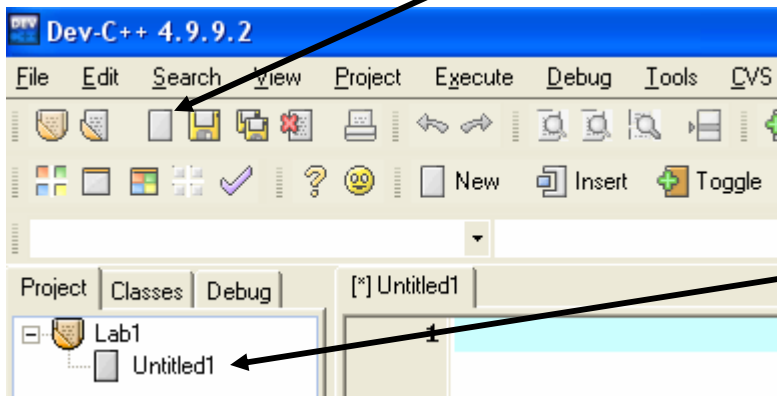
These tabs will be covered later in the course.

Using the DevC++ Editor

The DevC++ Editor window is where you will spend most of your time creating, editing, and navigating through source code for your project. This section introduces you to these editing tasks:

- Creating a New Text File
- Entering Text into a File
- Saving Changes to a file

click here to add a new text file to your project



You now have a new file in your project, however, it is not saved and does not have a name. For simplicity save it in the folder you created earlier on you desktop.

Creating a New Text File

To create a new text file, just choose **New** from the **File** menu. A new untitled editor window will appear, ready for you to enter, edit, and save text or source code. In most of this course's projects, we will begin by editing the existing stub source file created for you project in `main.cpp`. Later we will add other files containing user defined libraries, classes, documentation and implementation source files

Opening a File

There are a variety of ways to open a source code or text file from within a project. Just double-click its name in the Project window. You can also select a file in the Project window using the arrow keys and press the Enter key to open it. You can also open an editable text file within a group by using the group pop-up that appears to the right of the group name. To open a file from a pop-up menu select the file from a group pop-up in the Project window. The file opens and appears in its own window.

Editing Files

Writing programs consists largely of typing code into an editor. And just like a good word processor makes it much easier to write a paper, a good programmer's editor makes it much easier to write a program. You can edit several files at one time in the DevC++ C++ IDE. An editor is a pane looking window into which you type text. The DevC++ Editor operates like a regular text editor, with many special features. To enter text, position the cursor at the point where you want to insert text, click the mouse button to place the insertion point, then begin typing. Your keystrokes appear at the insertion point.

After you edit a file, you can easily save it:

1. Save the file. Select **File** → **Save**
2. Save a file, giving it a new name. Select **File** → **Save as**
3. Save any files that have changed. Select **File** → **Save all**

If you are familiar with any typical word processor, you will find that the DevC++ IDE has similar editing tasks like selecting, cutting, copying, and pasting text, moving around in the document and finding text. Color syntax highlighting makes it easier for you to read the program you write and helps you find common syntax mistakes quickly.

Building Projects with DevC++

In this section we'll introduce you to ways of building a project. Among the tasks the IDE provides for compiling, running, and debugging your project are:

- Compiling a Project
- Compiling a Current File in a Project
- Running a Project

- Debugging a Project
- Compiling and Running a Program

Note that these selections under the drop down menus also have speed bar icon buttons that do the same command directly without using the drop down menu.

Compiling a Project

Use **Compile** on the **Execute** menu to compile all modified files. This process creates object code for compiled files, but does not link the code into binary file. This does not run the application, but is like a “spell checker” looking for correct syntax.

Compiling Current File in a Project

Use **Compile Current File** on the **Execute** menu to compile a selected file. Checks syntax in the selected file, it does not interactively run the final application.

Running a Project

Use **Run** on the **Execute** menu to launch the application created by your project. If the project type is not an application, the **Run** command is disabled.

Debugging a Project

Use **Debug** on the **Debug** menu to launch the application under control of the Debugger. you can then debug your code.

Compile and Run a Project

Use **Compile and Run** on the **Execute** menu to check the syntax and if correct run the application.