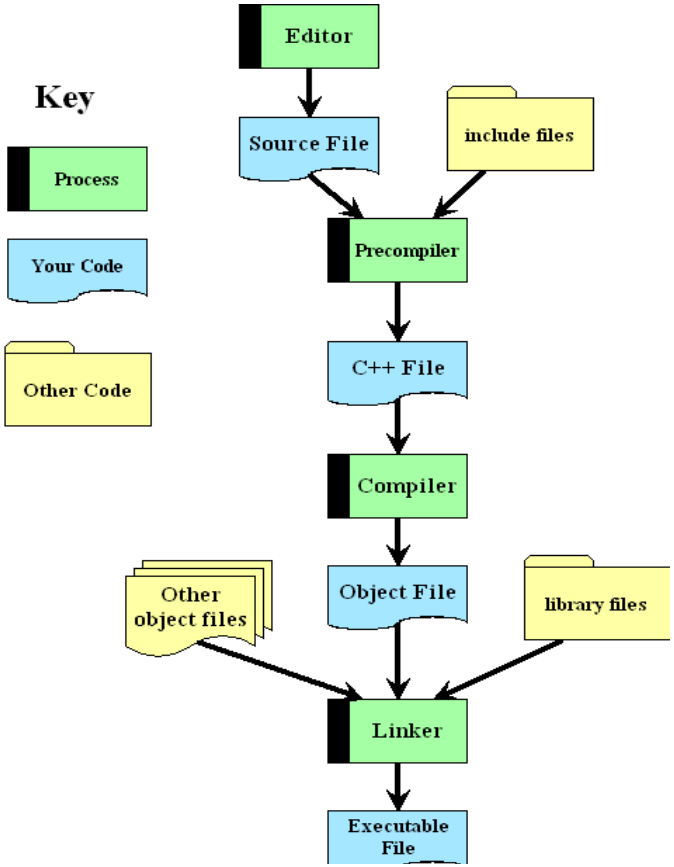
The Programming Environment

Nowadays, most code is developed inside what is known as an IDE—an *Integrated Development Environment*. An IDE integrates all the tools you need to prepare, debug, test and run a program.

Recall the process introduced earlier



Preparation of a program requires the following basic steps:

1. *Edit* your module to produce a *source file*
2. *Compile* your module to produce an *object file*
3. *Link* the object file with other *object modules* to produce an *executable file* (a program).
4. *Run* the executable (your program).

All of these steps can be done from within the IDE.

For convenience, steps 2 and 3 may be combined in which case they are referred to as a *build*.

Once a program has been built, it is a standalone program.

It may be executed independently from the IDE

Debugger

In addition to basic editing and build tools, IDEs will generally provide a *debugger* which will allow you to run your code interactively.

Setting up a Project

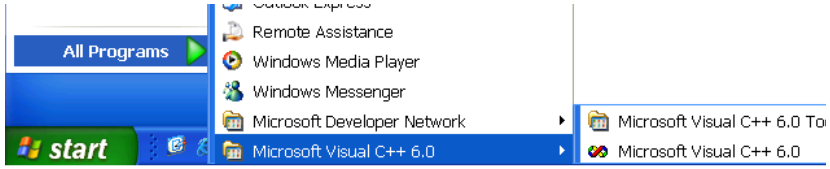
You might want to bookmark this page.

You need to understand not only what it is you are to do (what buttons to push) but what it is that your doing (the structure and function of the files and directories being created).

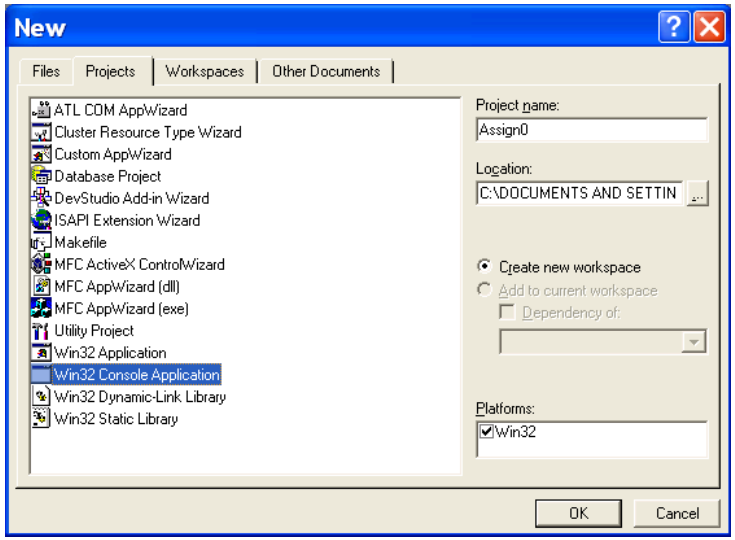
Step 1: Create a directory for the course (if you are in the lab do it on your M drive). Call it `programming` or `2420` or any other convenient name.

Step 2: Create a directory for your assignments, something like `assigns`.

Step 3: Now start up the `Visual C++` program. (How you do this will vary with your computer setup. On most machines, go to the `start` menu, select `programs`, then `Microsoft Visual C++ 6.0` which will give you another menu. Select `Microsoft Visual C++` from that one as well.) Here is how it looks in Windows XP



Step 4: Once the Visual C++ envuronment is open (and you've closed any hints boxes) choose `File` menu then pick `New`. You should see the following dialog box:



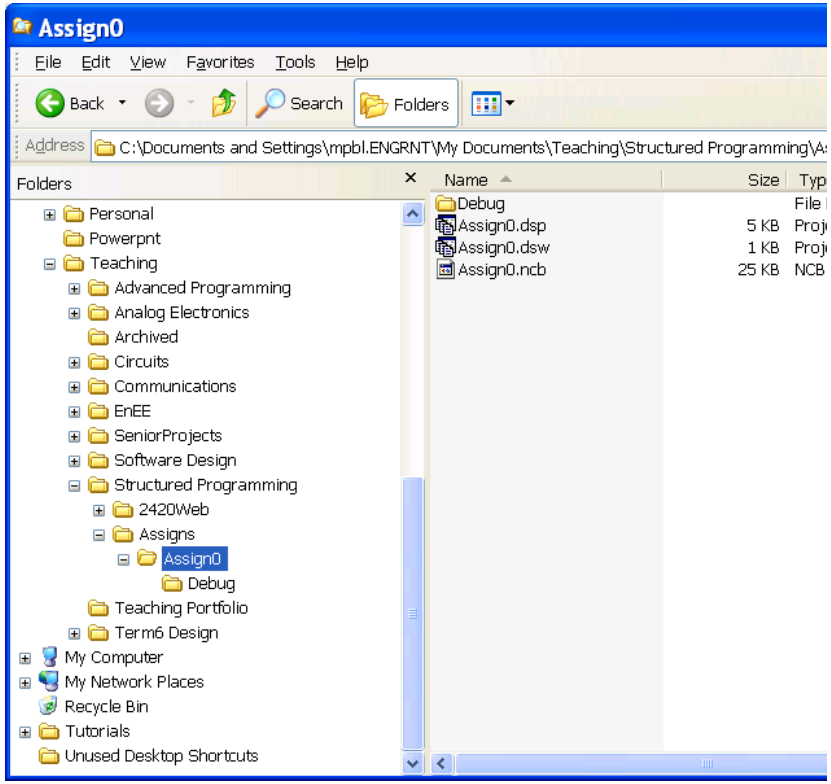
Here you need to do three things:

1. Choose the `Win32 Console Application` as shown.
2. Pick the `...` box to the right of the `Location:` box and navigate to your `Assigns` directory.
3. Then type the name, `Assign0`, into the project box for you practice assignment.

Once all your choices have been made, hit `OK`.

At this point what you have done is created a Project Workspace, a place where all your project files will be kept.

This includes special files required to manage the project. Let's use Windows Explorer to see what the IDE has done.



Here, in the `Folders` view on the left, we can see the `Assigns` directory was created underneath the `Structured Programming` directory by the programmer.

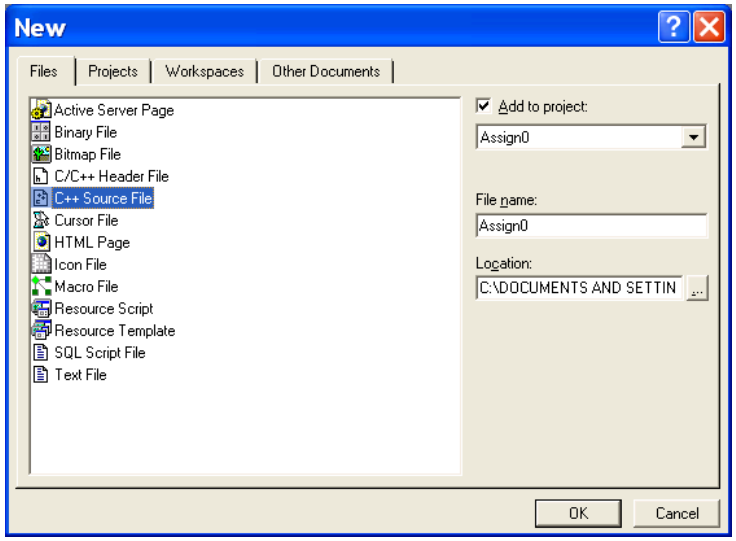
Then *the IDE created* the `Assign0` directory under that and populated it with all the files we see in the right pane.

The `Assign0.dsp`, the `Assign0.dsw` and the `Assign0.ncb` files are all project management files that the IDE creates.

It also created the `Debug` directory (which so far is empty).

Creating A Program Module

Step 5: Once again choose `File` menu then pick `New`. The dialog box should look a little different this time:



It's actually the same dialog box but the `Files` tab should have been automatically selected this time. The IDE knows you've already created a project so it assumes you want to create one of the modules (files) for the project.

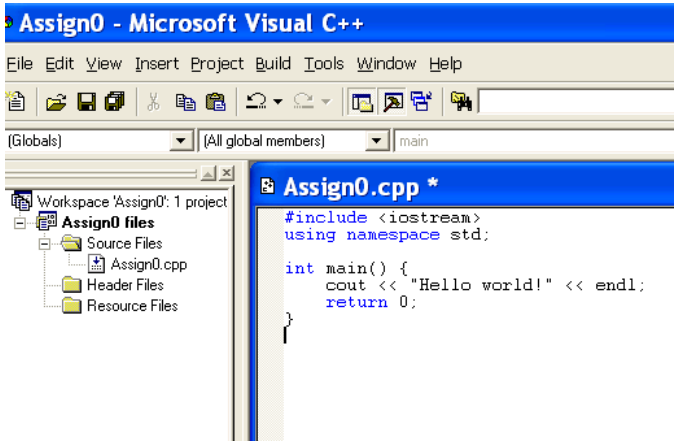
You can over-ride the choice by selecting another tab. But this is the one we want now.

1. Choose `C++ Source` File
2. Make sure `Add to project` box is checked
3. `Assign0` should already appear in the `Project` box
4. Type the name of the file in the `File Name` box. We'll use `Assign0` again. (Projects usually contain many C++ files but the one with `main()` in it is usually named after the project.)

Hit `OK`.

The IDE will create an empty file for you.

Step 6: Type in your program module. In this image, we've created a very basic Hello world program.

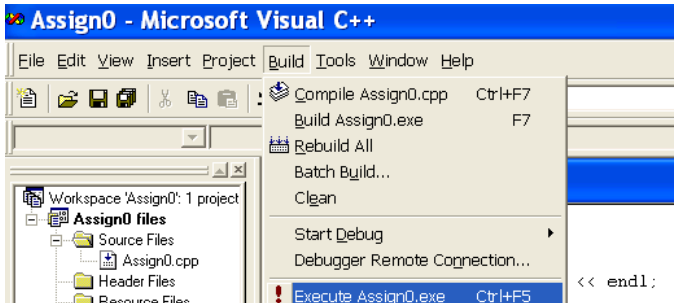


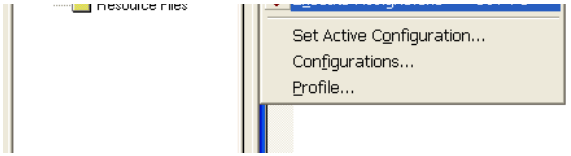
In the left hand pane we have selected the `Files` tab which will be more usefule to you this term than the `classes` tab.

It shows that there is a single source file in the `Assign0` project files, called `Assign0.cpp`.

The right window shows the simple code typed into the actual file. The `*` in the titlebar after the `Assign0.cpp` is telling you that the file has changed since it was saved last. When we save the file it will clear.

Step 7: Use the `Build` menu to compile, link, build (do both together) or run our file.

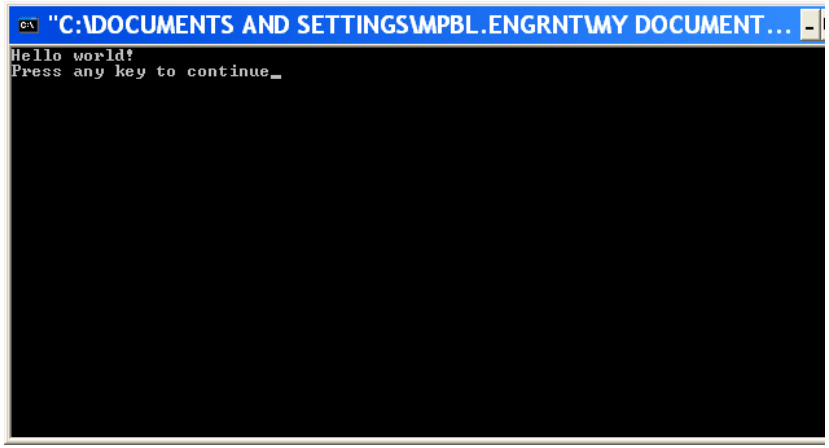




Here we have highlighted execute `Assign0.exe`. When we select this, we get a dialog that informs us we haven't built the project yet and asking us if we want to do so. Select `OK` or `yes` and the following will happen:

1. The file will be saved
2. It will be compiled.
3. If there are no grammar errors the resultant `Assign0.obj` file will be linked with the `iostream` library.
4. If there are no linker errors, the resultant `Assign0.exe` file will be executed.

The file is executed inside a windows command console, a small, plain window that emulates an old fashioned DOS screen

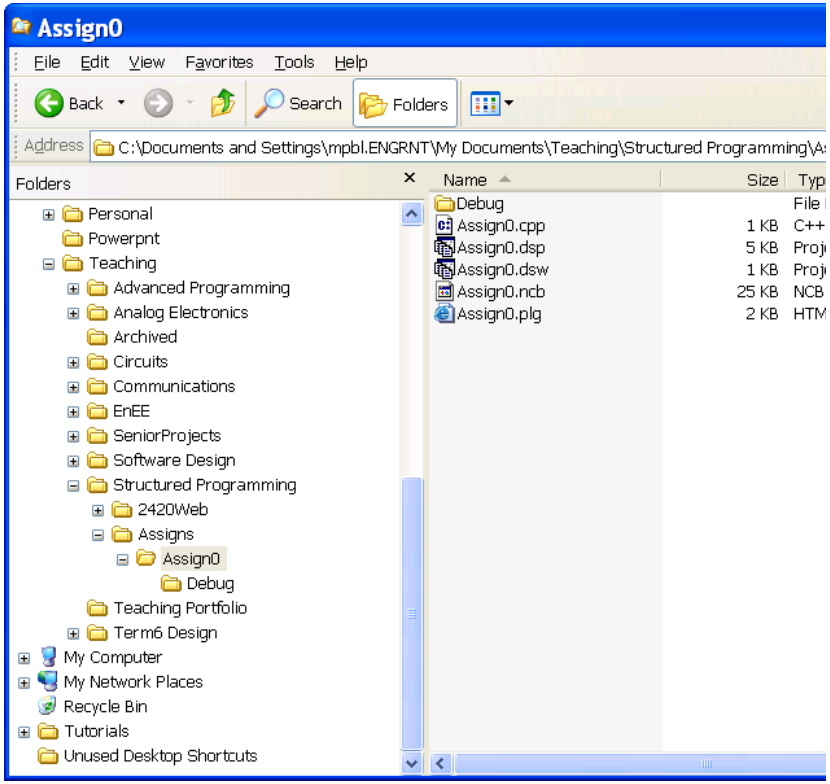


Once your program has finished, you will see a line saying `Press any key to continue`. Since our program only outputs one line and then ends, we see the `Hello world!` line first.

Hitting any key will close the console window.

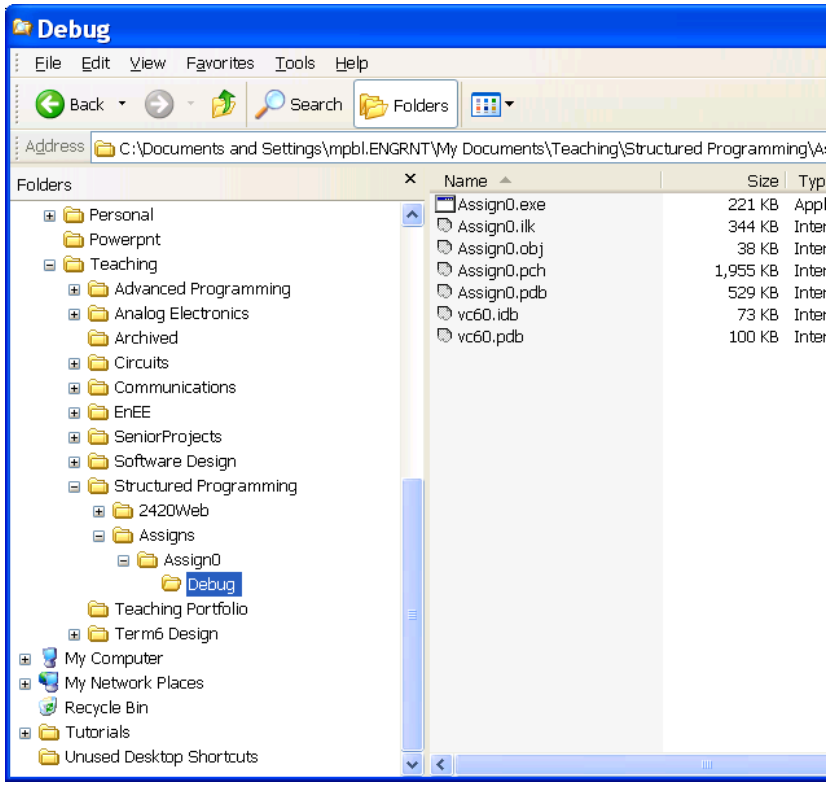
Cleaning Up After

When you're done developing your program you will find the project directory file has grown



Not only has our `Assign0.cpp` file been added, so has the `Assign0.plg` file. No, we don't know what it is exactly either. Yet another maintenance file added by Microsoft.

It gets worse. Check our formerly empty `Debug` directory



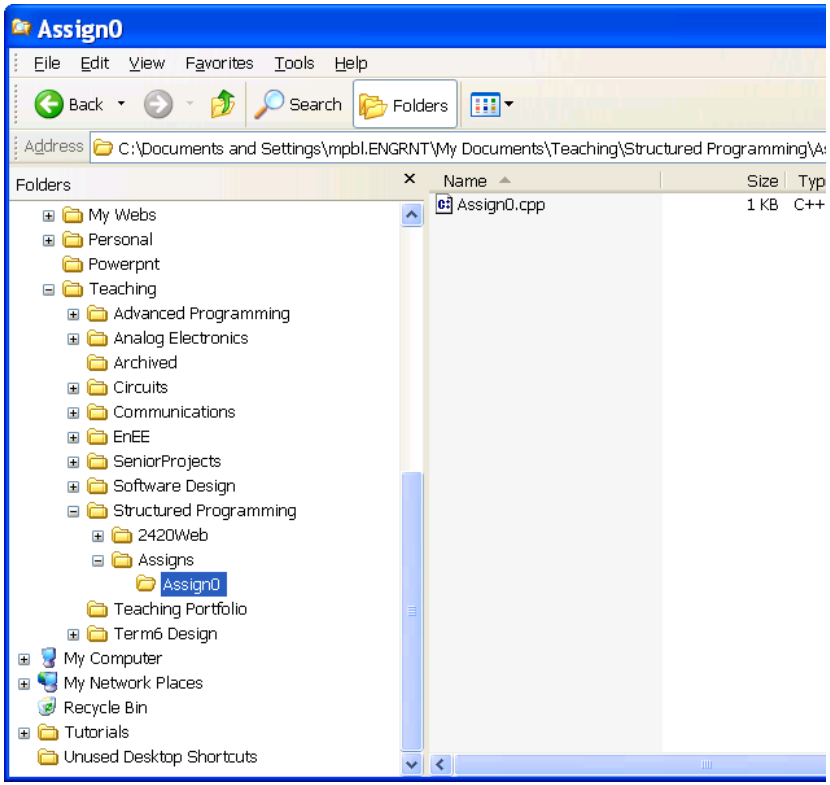
and you'll find a whole bunch more files, one of which is almost 2MB in size!

The one you care about is `Assign0.exe`. This file can be copied, sent to your Mom, sold to the highest bidder. It is your stand-alone executable file. Doubleclick on it and it will run (inside a command console again).

Step 9: When you are done with an assignment, all files *except your source file(s)* should be erased. They take up a lot of space on your M drive. Don't worry. If you need them they are easy to re-create.

Make sure you close the Visual C++ IDE first or it may not let you erase some files.

When you're done, this is all you should have:



Interactive and Batch Mode

Interactive Mode
: A (human) user will enter the data in response to prompts while the program executes.

Batch Mode
: Runs without direct human interaction. Data is read from a file, and output is written to a file.

Note: The assignment testing is done in batch mode.

Input/Output Redirection

On Unix, Cygwin, or in a command window:

`myprog < mydata.txt` —run myprog using `mydata.txt` for the standard input (`cin`).

`myprog > myoutput.txt`—run myprog using `myoutput.txt` for the standard output (`cout`).

`myprog < mydata.txt > myoutput.txt` —run myprog using `mydata.txt` for the standard input, and `myoutput.txt` for the standard output.

Programming Errors

Syntax Errors
The program code contains something that isn't a valid C++ statement.
- Detected by the compiler.
- Fix the first one first—a simple typo can result in many error messages.

Run-time errors
The program attempts to perform an illegal operation (e.g., divide by zero).

Undetected errors
The program runs but doesn't give the right results. Be very careful to check that the results are what you expect.

Logic errors
The algorithm is incorrect. Desk check (walk through) your program.

Recreating a Project

If you need to recreate your project later on, first make sure you have your (in our example `Assign0.cpp`) file and it is still in the `Assign0` directory. Then follow **Steps 3 & 4** above except that in Dialog Box for creating a new project (**Step 4**) select the directory which contains your project directory as the location (in this example the `Assigns` directory) then enter the name of your existing assignment directory as the project name (e.g. `Assign0`).

Instead of **Step 5** (creating a new file) click on the `File` menu, select `Open`, and open your existing file. Then right click on your opened file and select `Add to Project` and voila! you're back in business.

This page last updated on Thursday, January 8, 2004