

Quick Introduction


The following pages provide a quick tutorial on using Microsoft Visual Studio Express 2008 C++ to produce a small project.



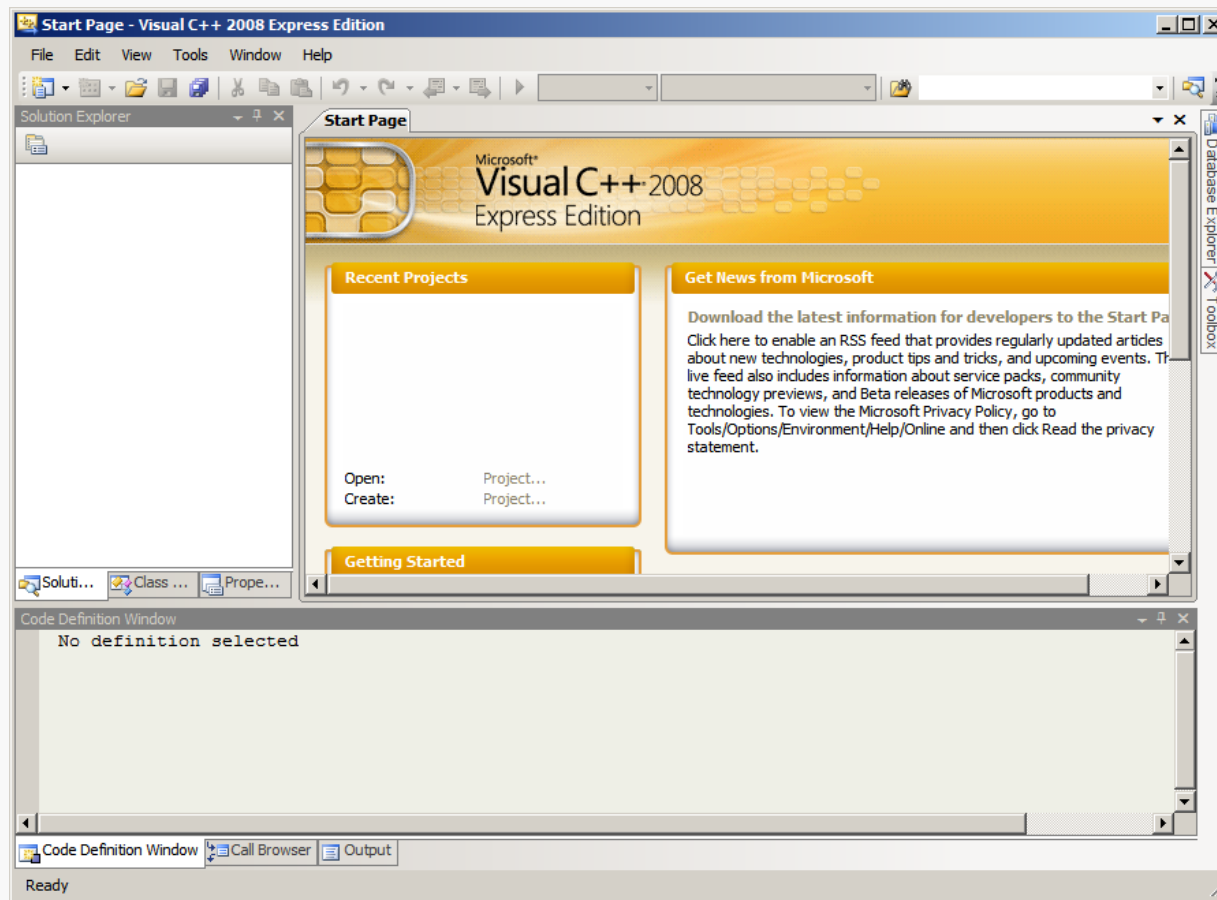
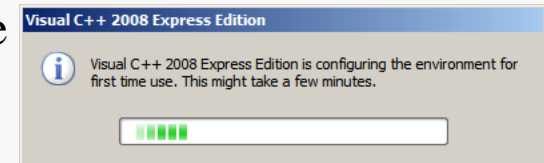
The following discussion is based on an actual session using the Visual C++ Studio Express 2008. The menu selections and options illustrated here do not conform to earlier versions of Visual C++.

A student may choose to use another existing ANSI standard C++ compiler/IDE if they wish. However, it is the student's responsibility to ensure that their programs compile and run under the Microsoft Visual Studio 2008 Express C++ environment. TAs will only be supporting the Microsoft Visual Studio 2008 Express C++ IDE. This means that students who choose to use other compilers cannot expect the TAs to help them with specific compiler problems, (e.g., interface questions, compiler messages, warnings or errors). Instructors and TAs will be prepared to help students that use the required compiler. Students that use another compiler do so at their own risk.

Getting Started: MS Visual C++

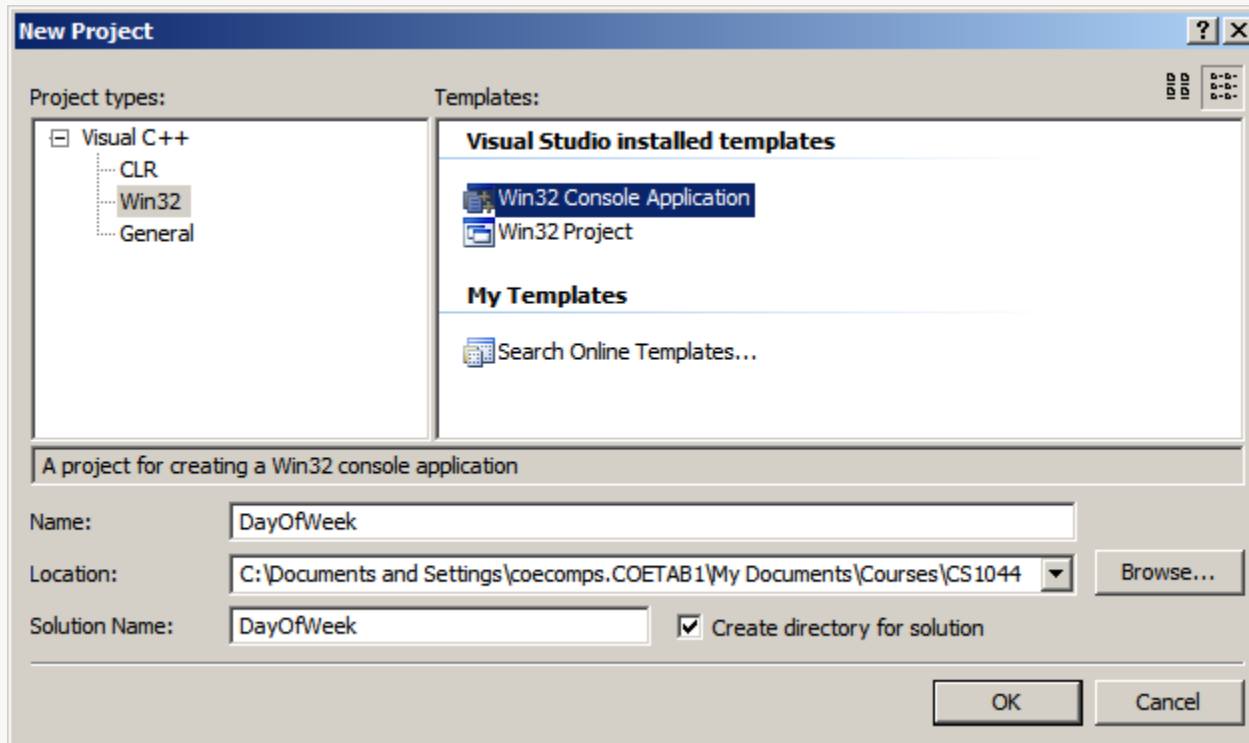
To start Microsoft Visual C++, click on  Start, select Programs > Visual C++ 9.0 Express Edition > Microsoft Visual C++ 2008 Express Edition. (The first time you execute it would will get the Window at the right.)

You should now see a screen that looks similar to this:

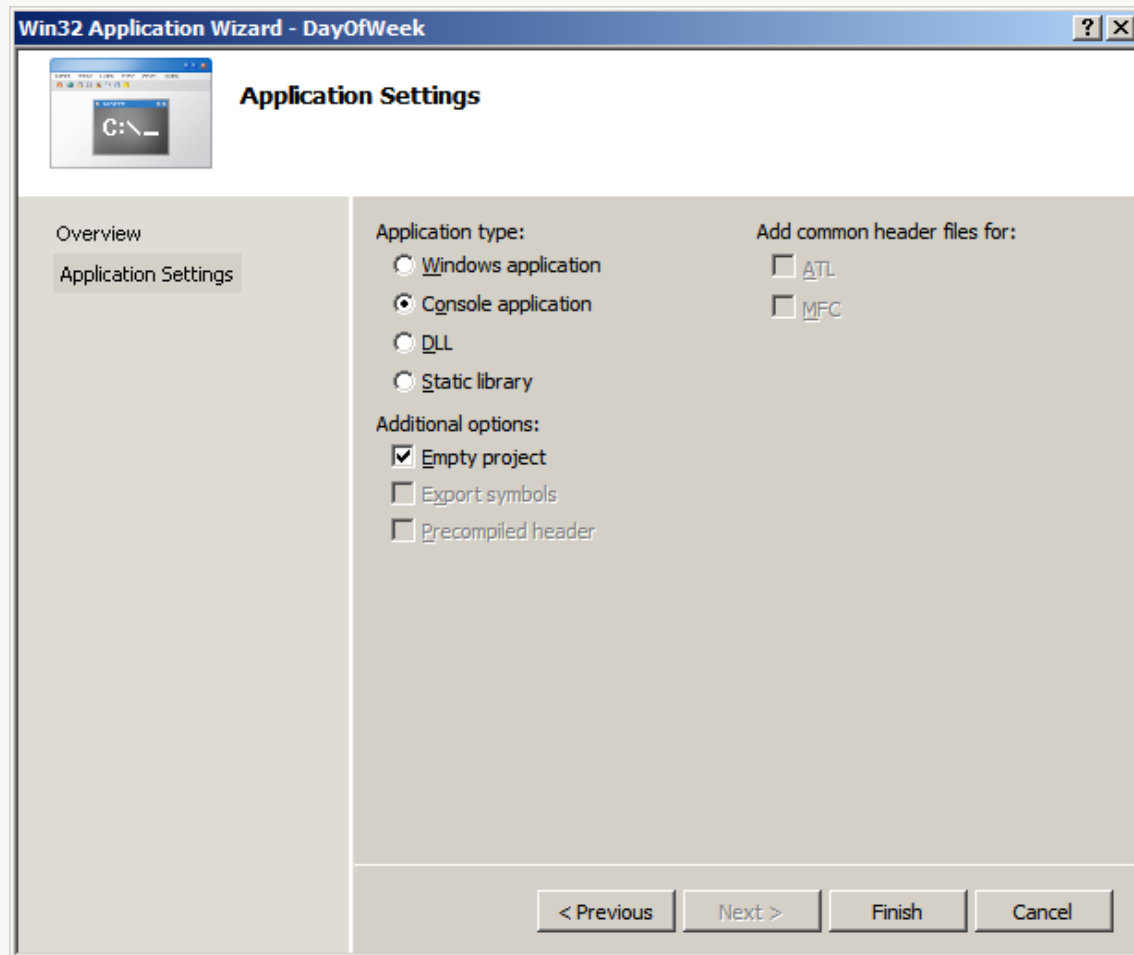


Before you can begin coding, you must create a new project.

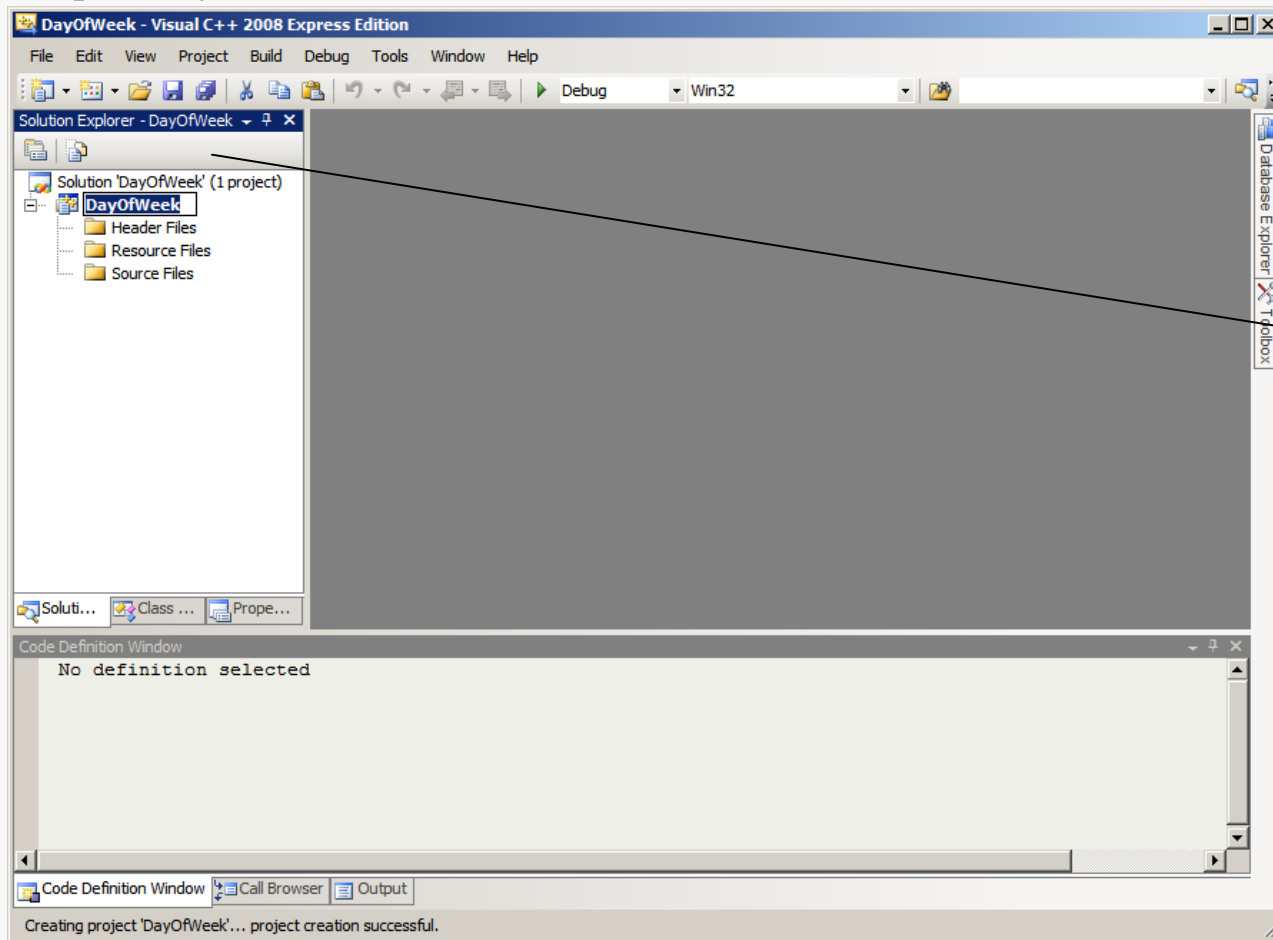
To do so, go to the File menu and select New ► Project... (or click on the New Project button below the file menu). You should see a dialog similar to the one below. Note that the Location of the project will be where a new directory is automatically created to hold all of the files for the project. Be sure to select Win32 project type and the correct project template: Win32 Console Project:



After clicking Ok from the New project dialog window you will be presented with the project creation wizard “Win32 Console Application” dialog window as shown below. Click on Application settings & confirm the settings as depicted below then click Finish.

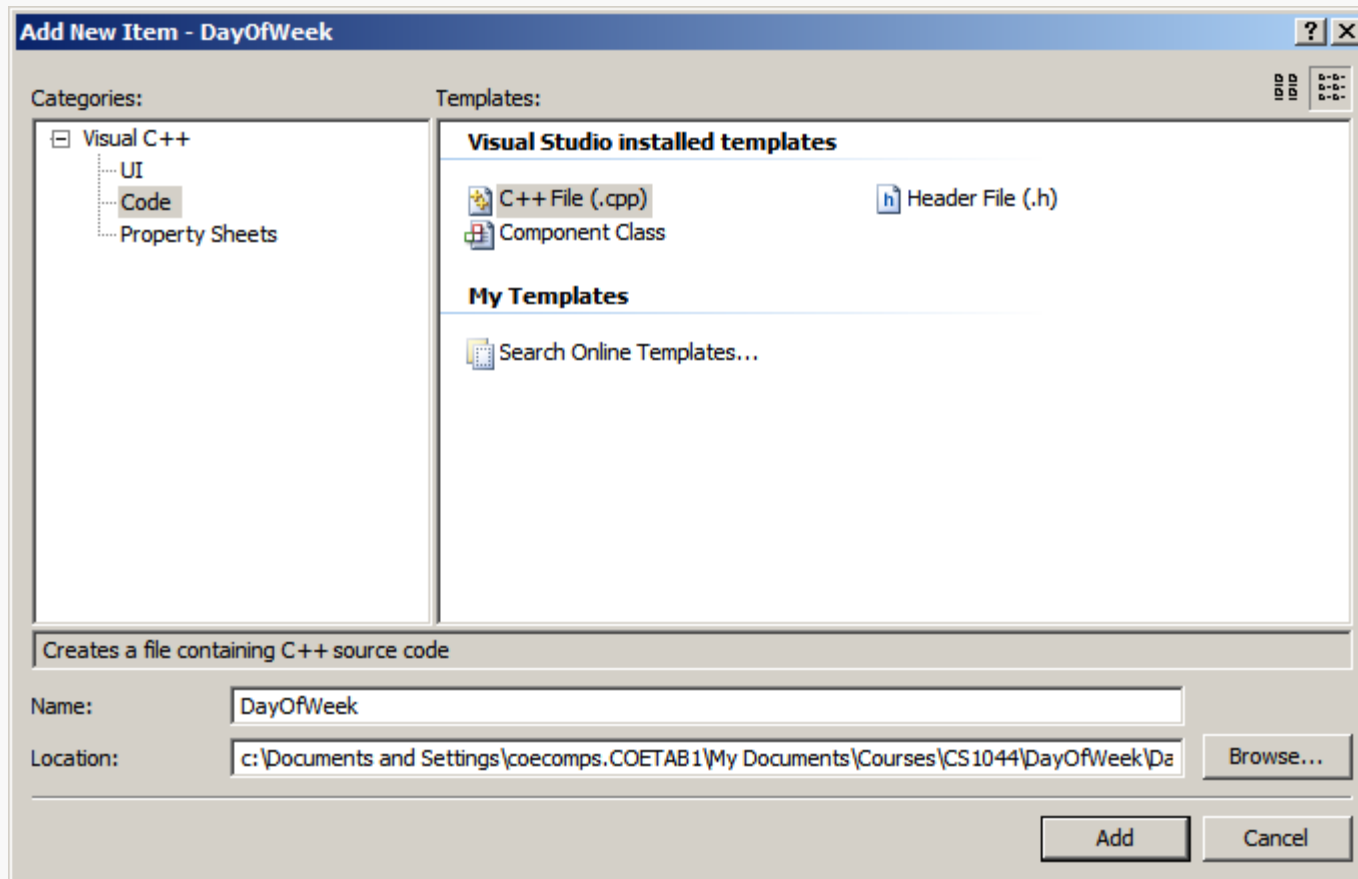


If you selected Win32 Console Application (this will create an application that runs at the command prompt [like DOS]). In the Project Name text field, if you named your project DayOfWeek. In the Location text field, you typed the directory in which you wished to store your code files. After clicking OK, Close the Start Page within MS Visual Studio and select View menu / Solution Explorer, you should see a screen that looks similar to this:



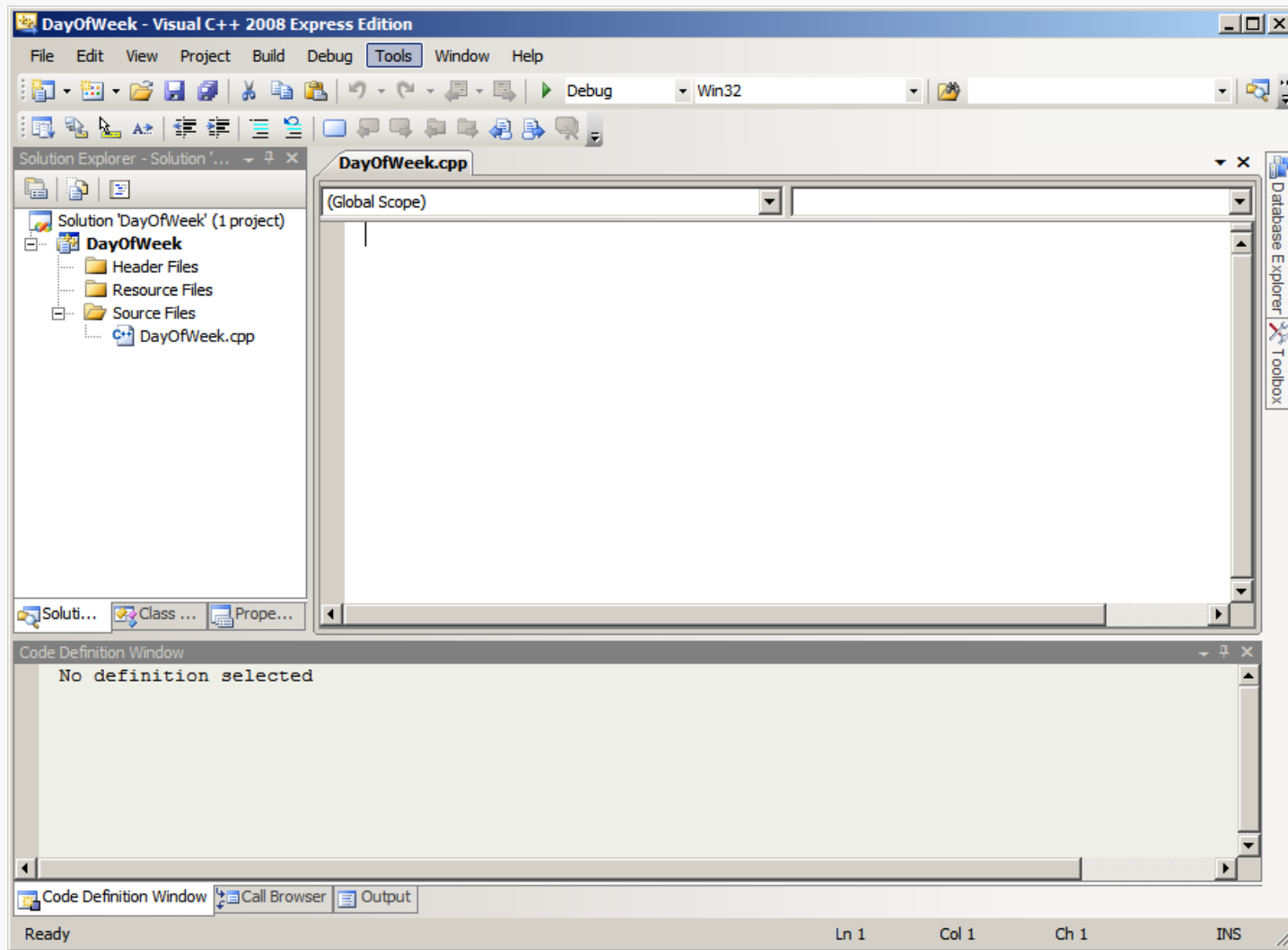
If you don't see the Solution Explorer pane, pull down the View menu & select Solution Explorer.

Now that you have created a Project, you can add source code to it. To do so, go to the Project menu, select Add New Item... You should see the following: Select Code / C++ File, enter DayOfWeek for the name and click on Add to continue.



Project + Source Code File:

You should now see a screen that looks similar to this:



Day of Week Program

Now in the edit window that just appeared type in the following C++ source code:

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main() //Find the Day of the Week for a Date
{
    int    mon, day, year;

    cout << "Enter a date for which you wish to know" << endl;
    cout << "the day of the week (MM DD YYYY)? ";
    cin  << month >> day >> year;

    if (year < 1752)
        cout << "Only Gregorian dates accepted, sorry " << endl;
    else {

        if (month < 3) { //Jan & Feb = 13 & 14 preceding year
            month += 12;
            year  -= 1;
        } // end if
```

Day of Week Program

```
    weekDay = (day + 2*month + 3*(month+1)/5 + year +
               year/4 - year/100 + year/400 + 1) % 7;

    if (month > 12) { //reset Jan & Feb
        month -= 12;
        year  += 1 ;
    } // end if

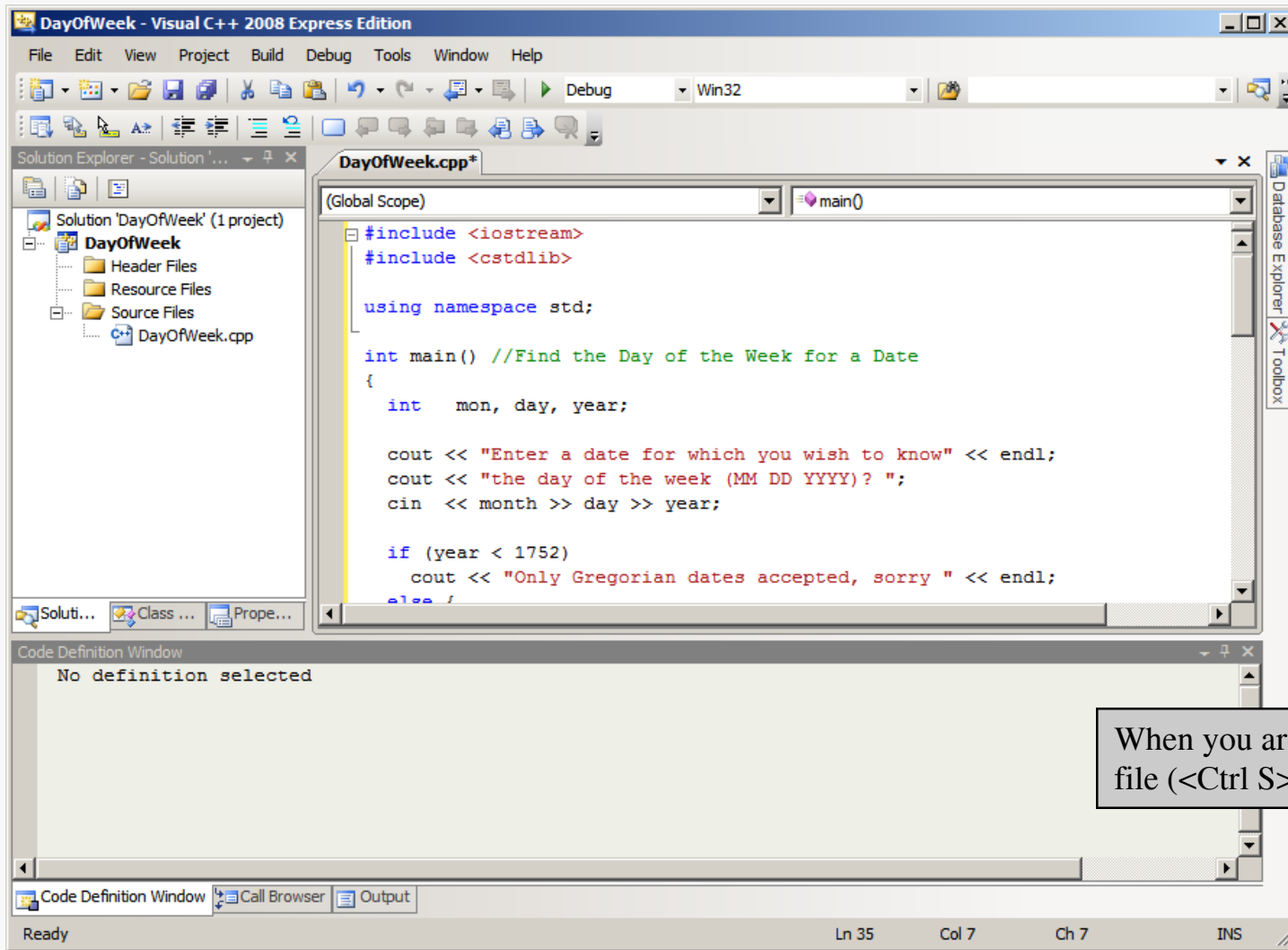
    cout << month << "/" < day << "/" << year << " falls on ";
    switch (weekDay) {
        case 0: cout << "Sunday" << endl; break;
        case 1: cout << "Monday" << endl; break;
        case 2: cout << "Tuesday" << endl; break;
        case 3: cout << "Wednesday" << endl; break;
        case 4: cout << "Thursday" << endl; break;
        case 5: cout << "Friday" << endl; break;
        case 6: cout << "Saturday" << endl; break;
    } // end switch

} // end else

return EXIT_SUCCESS;
} // end main
```

Code Entry

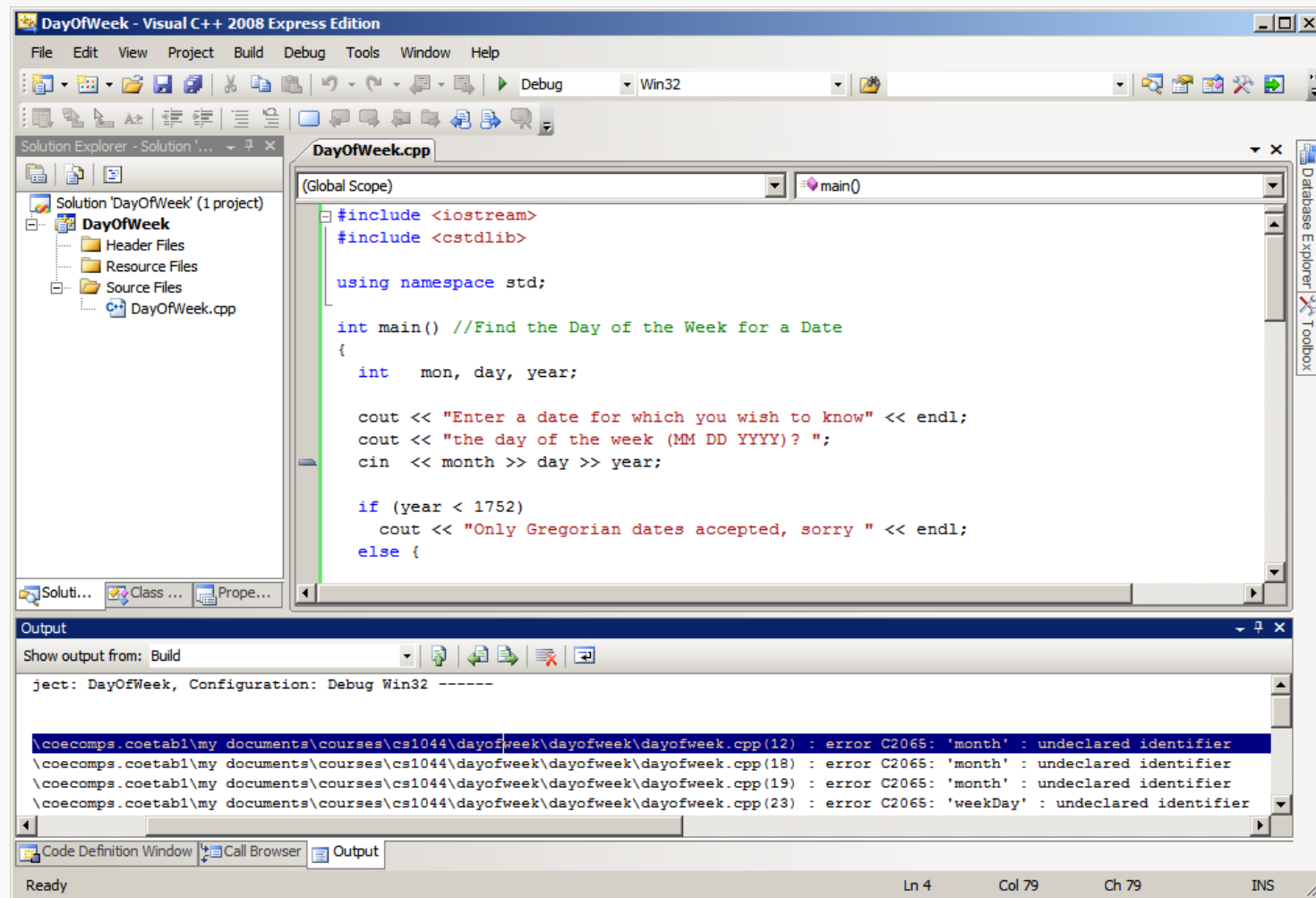
Now that you have entered all of your source code, you should have a window that looks like:



When you are done, save the file (<Ctrl S> or File, Save)

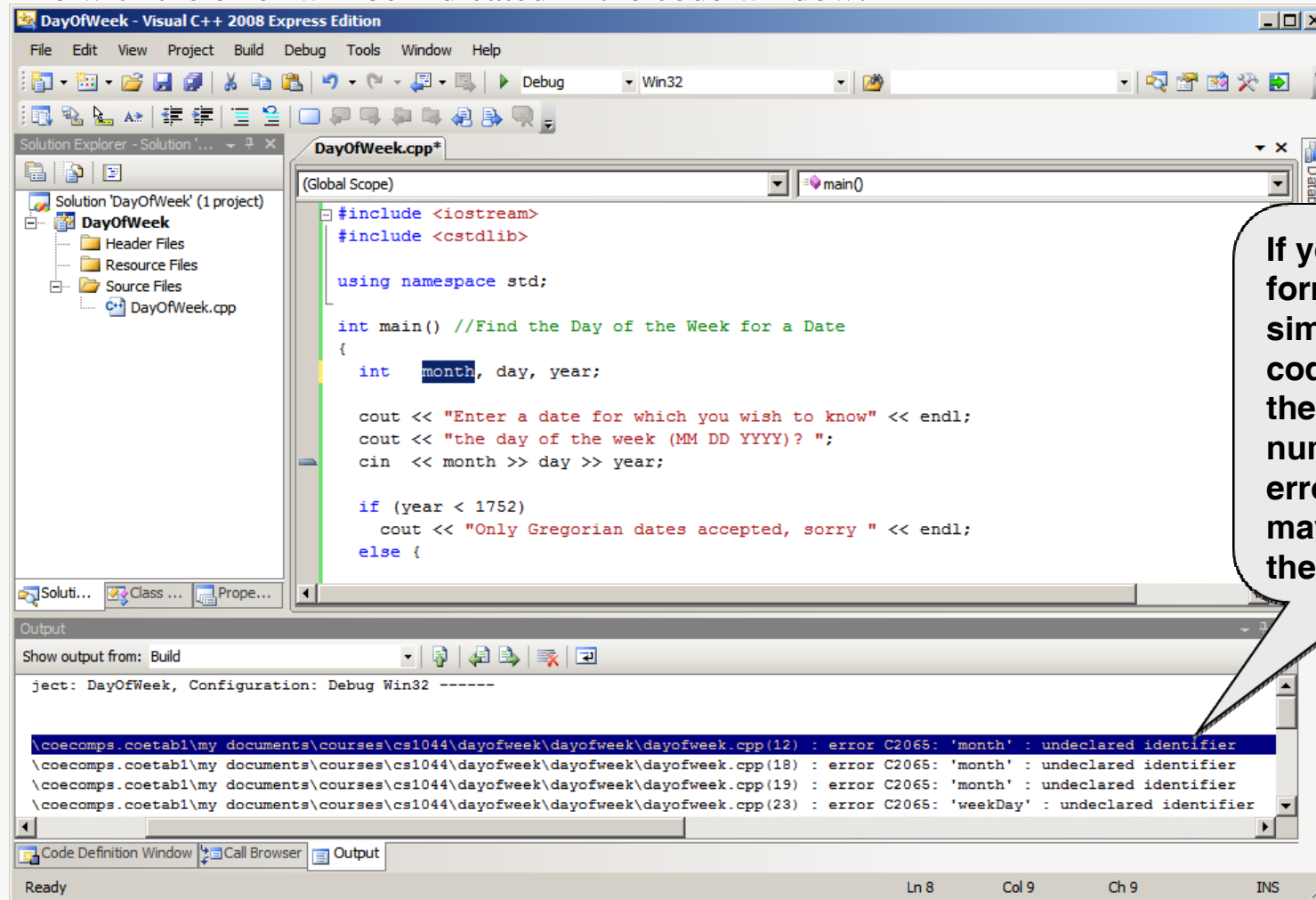
Compilation

You can now attempt to build your project. Building involves compiling and linking into an executable. Attempt to build the project, by hitting F7 or from the Build menu, select Build Solution:



Finding Errors

Notice the compiler error messages in the lower window. Double click on an error message, and the line with the error will be indicated in the code window:



Compilation Errors

Now correct the errors relating to each line and build the project again:

Error corrections:

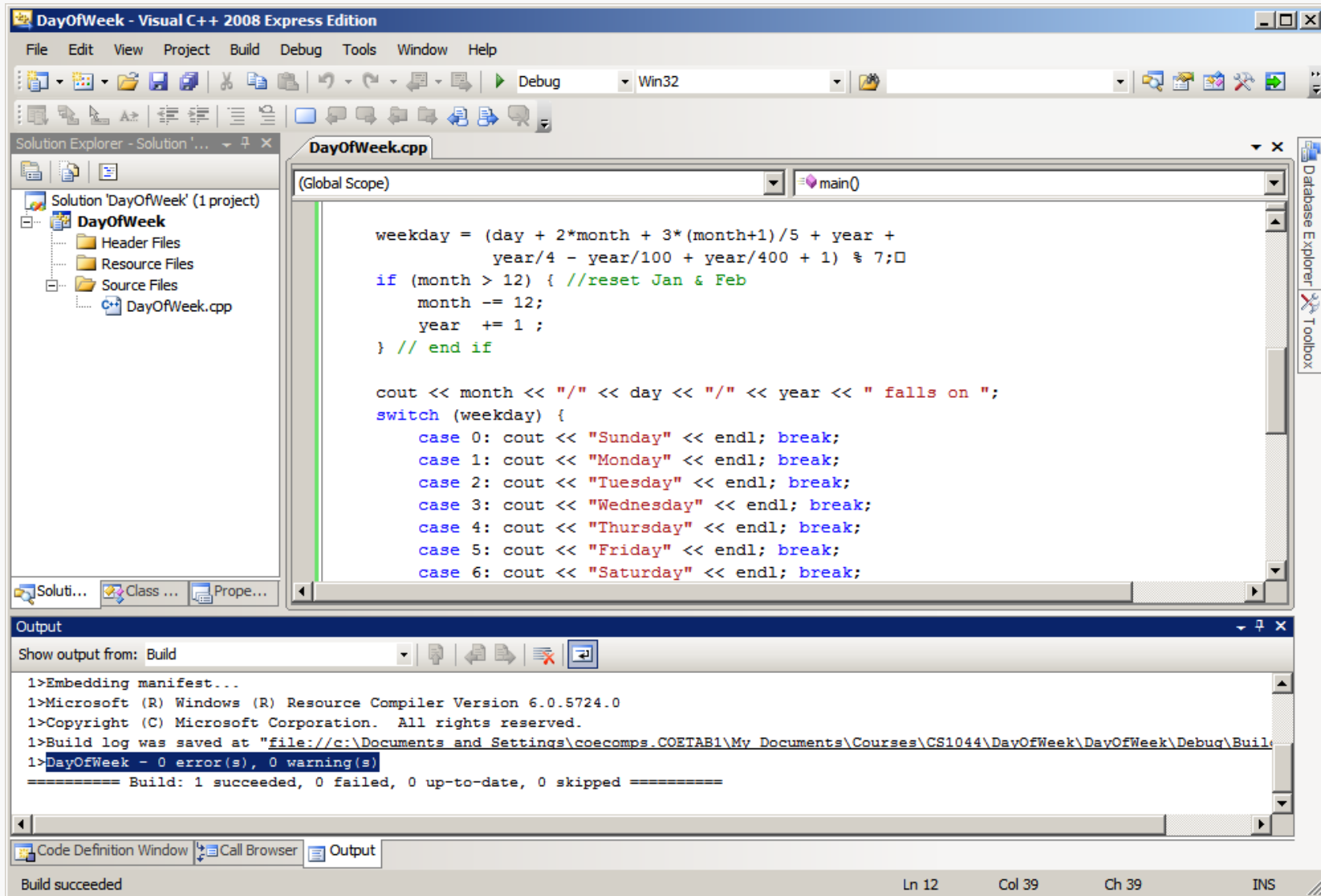
In case you can't figure out the errors here they are:

- Line 9: change mon to month
 add the variable weekday
- Line 13: change the << to >>
 (we want to save the value to a variable, not write it to the output stream)
- Line 22: change the { to } (we want to end the block, not start one)
- Line 24: change weekDay to weekday to match the declaration
- Line 32 change the < to <<

Continue correcting errors and rebuilding until you have no errors:

If you did not format your text similar to the code shown previously then the line numbers on the error messages may not match the ones shown.

No Errors



Program Execution

Now that your code has been compiled into an executable, it is ready to be executed. You can do so by pressing <Ctrl F5> or going to the Debug menu and selecting Start Without Debugging. Now you should see a DOS prompt with the program running:

