EEE101 C Programming and SW Engineering 1 Week 1 Lab Session – Getting Started with Compilers

Objectives

The main objective of this tutorial is to familiarise yourself with the process of writing and compiling simple C programs using the Microsoft Visual Studio 2013 command line compiler and the full Integrated Development Environment (IDE).
At the end of this tutorial you should be able to enter a simple C program which can be compiled and run.

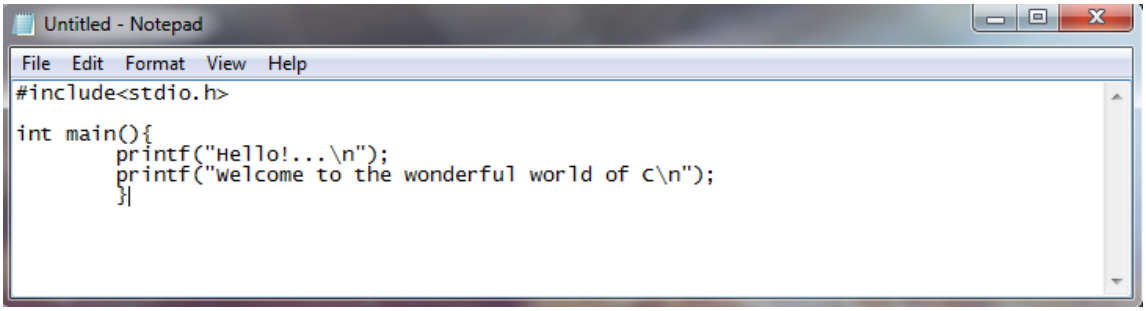
Introduction

For this laboratory, you will be using the Microsoft VS 2013 C++ compiler. This has been chosen as it is the development tool you will be using next semester in C++ also. However, rather than using the full IDE straight away, you will begin by only using the command line compiler & linker, writing your C programs using Notepad.
If you wish, you may choose to use another compiler e.g. Quincy 2005 (a simple to use freeware compiler)). **Note: Any coursework submitted MUST run in Visual Studio 2013 and MUST be written in standard C.**

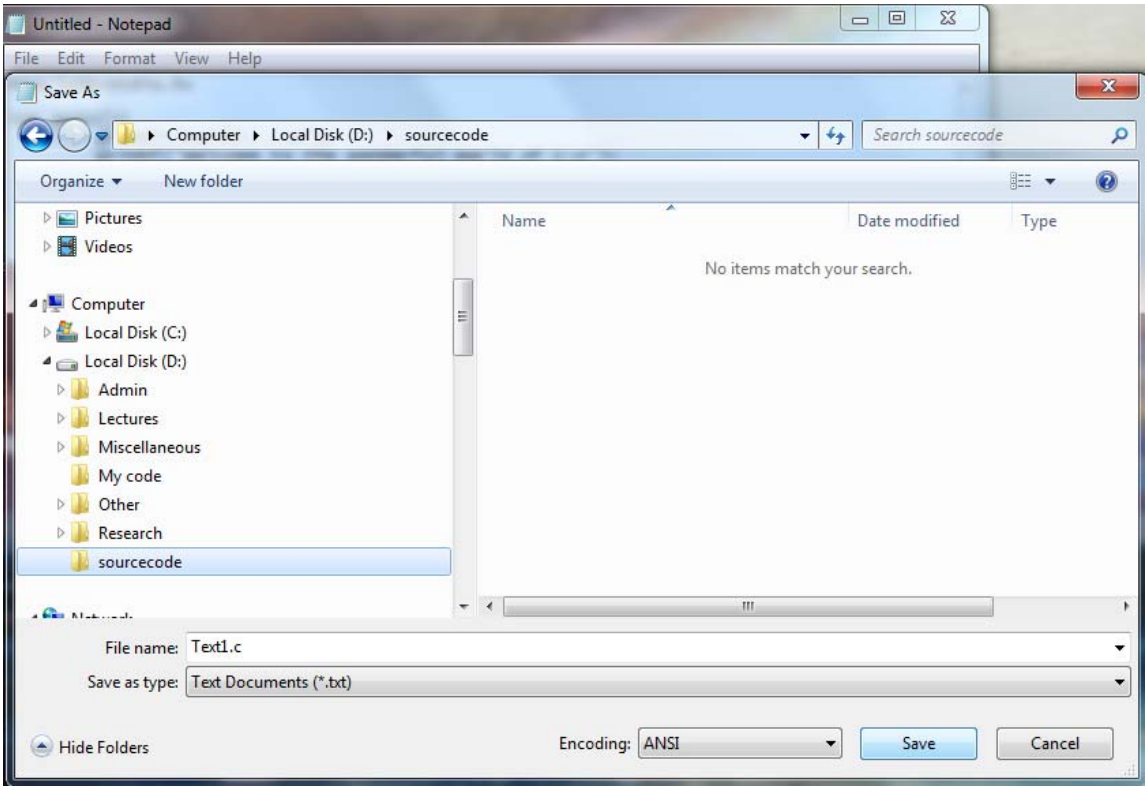
Typing and saving a simple C program

To avoid mouse clicks, you can just press: [Windows button]+[R], and then type "notepad", followed by [enter] (Or open notepad through start>all programs>accessories>notepad.

Then, type in the program shown in the figure below. Make sure you type it correctly!



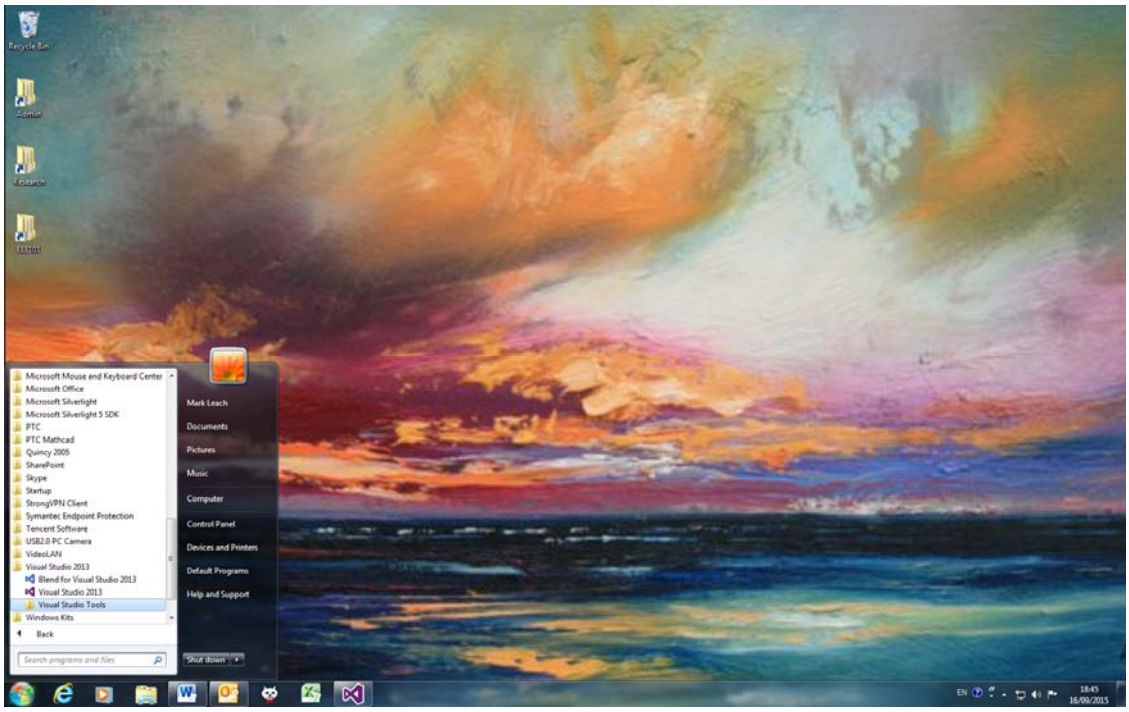
Create a folder called "sourcecode" in an easy to find place such as D:\sourcecode\, choose a file name that has **no spaces** and append "**.c**" e.g. Text1.c (see below)



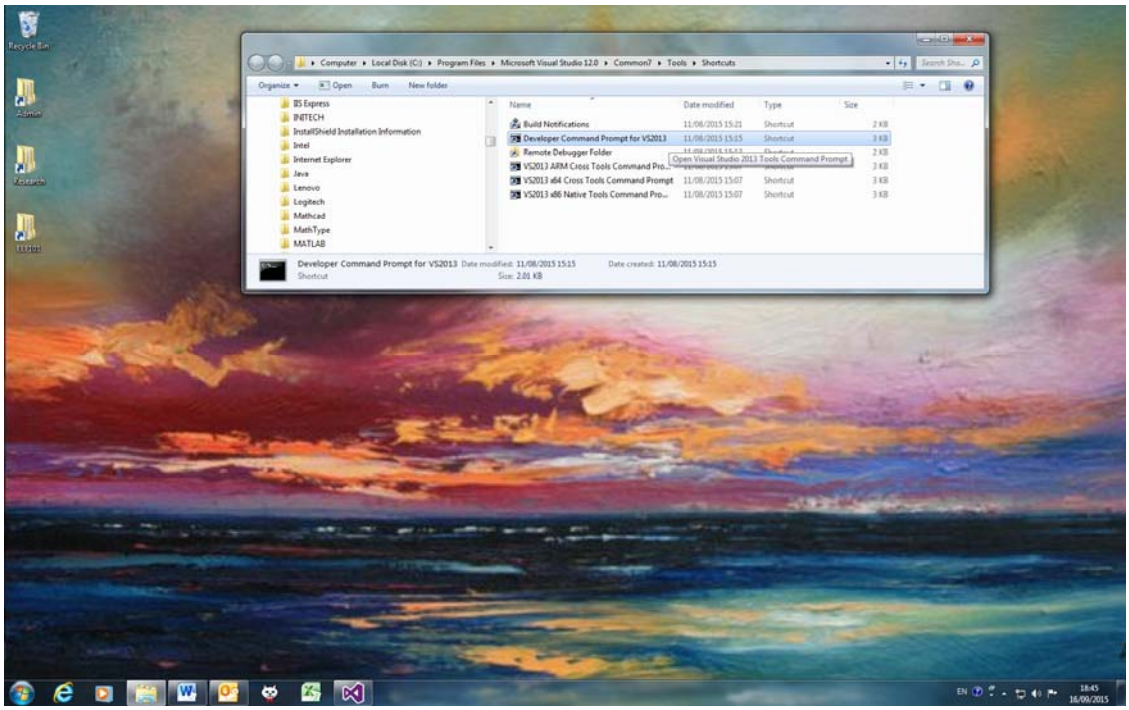
That is your C program written, now we need to compile it and see what happens when we run it!

Compiling and running your C program

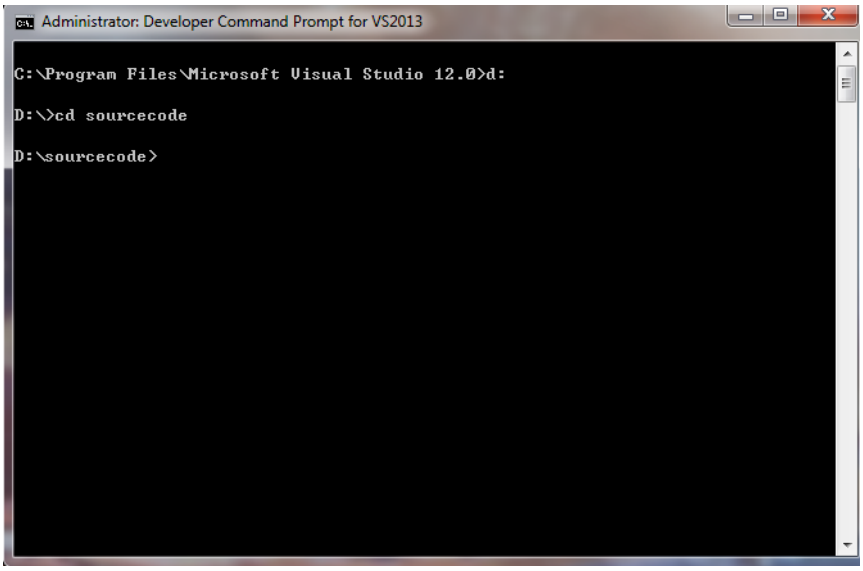
Open a Microsoft Visual Studio 2013 Command Prompt window that enables you to compile your program, by clicking the following menu sequence Start>All Programs> Visual Studio 2013>Visual Studio Tools (See below).



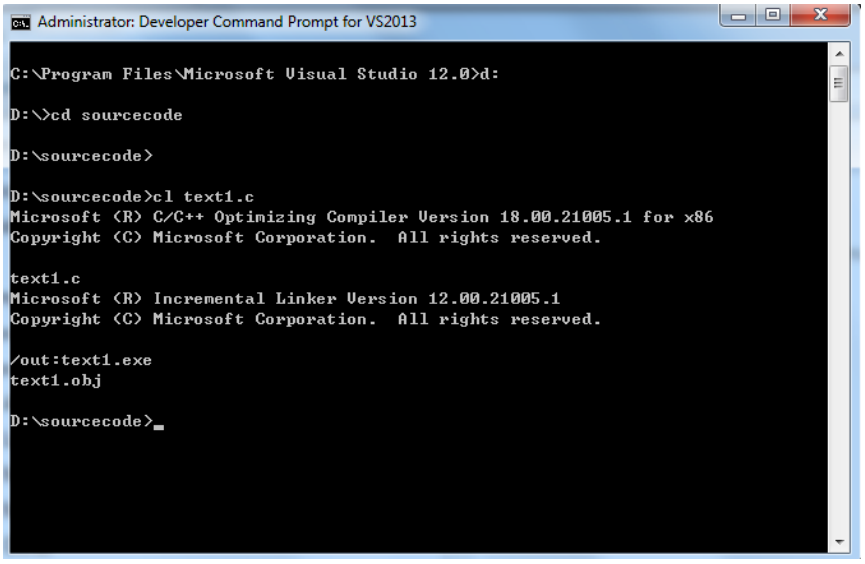
From the explorer window that opens, run “Developer Command Prompt for VS2013” (See below)



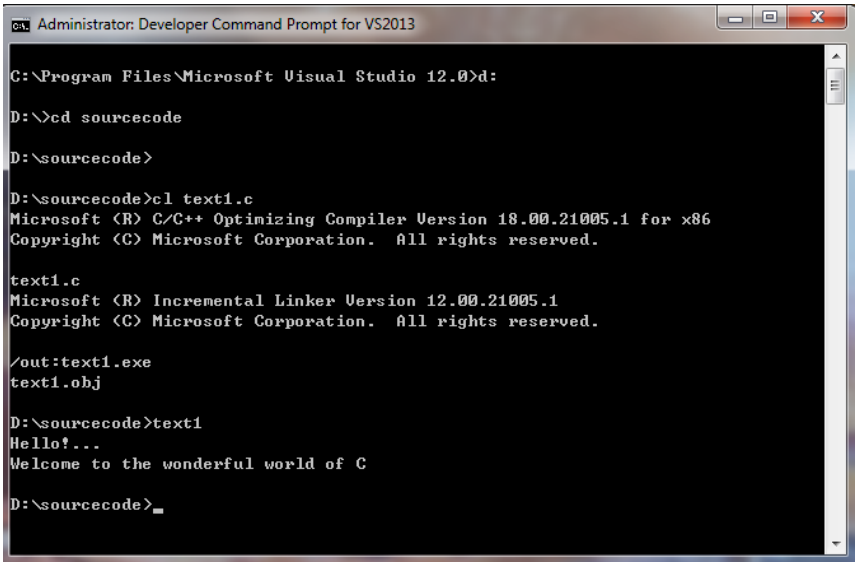
After the command window opens (see below), change the current directory to the directory where you have saved your file e.g. D:\sourcecode\. This is done by firstly changing the current drive type the drive letter and a colon d: and press enter. Then open your folder typing cd foldername (e.g. cd sourcecode) and press enter (see below).



To compile your program use the command cl (note that is an L not a 1) followed by a space and the name of your program file e.g. cl text1.c and press enter (see below).



The window shows that two output files have been created, text1.exe and text1.obj. The obj (or object) file is part of the building process and the exe (or executable file) is your executable program. To run your program type the name of the program (with or without the .exe extension) followed by enter. If you have not typed the program correctly the compilation will fail and produce errors instead of the two files, then you should review your notepad file and try again. The program should output on your screen the message you have instructed the computer to display.



More Information

The two files .obj and .exe have been created because, when calling the compiler application cl.exe you have used two applications. The actual compiler and another application called the linker.

The compiler has translated the high level C language instructions that you have typed in Notepad, which are contained in your file text1.c (this is also known as the source file), into the computer's machine code that the microprocessor can understand.

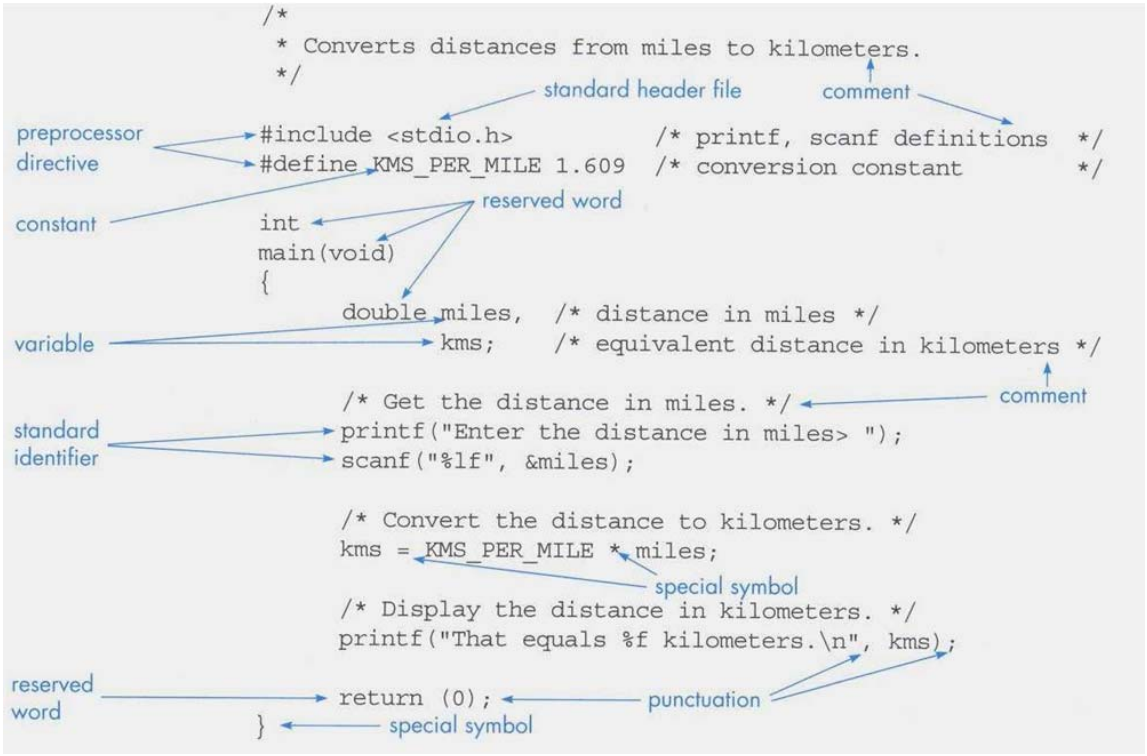
The first output of the compiler is the.obj. This file contains the machine language instructions that carry out the program's purpose. This file's format is binary so you will not be able to understand it if you open it.

Although an object file contains machine instructions, not all of the instructions are complete. High-level languages (like C) provide the software developer with readymade code for operations that the developer will likely need (e.g. printing on the screen). These pieces of code exist in header files (e.g. stdio.h). The linker, links the code in the header file with the functions in the object file to create the exe file ready to run.

To run the program, a loader provided by the operating system (OS) must copy (from the physical file in the hard drive) all its instructions into the memory and direct the CPU to begin execution from the first instruction. As the program executes, it takes input data from one or more sources and sends results to your display and/or secondary storage devices.

Compiling and running a more advanced program.

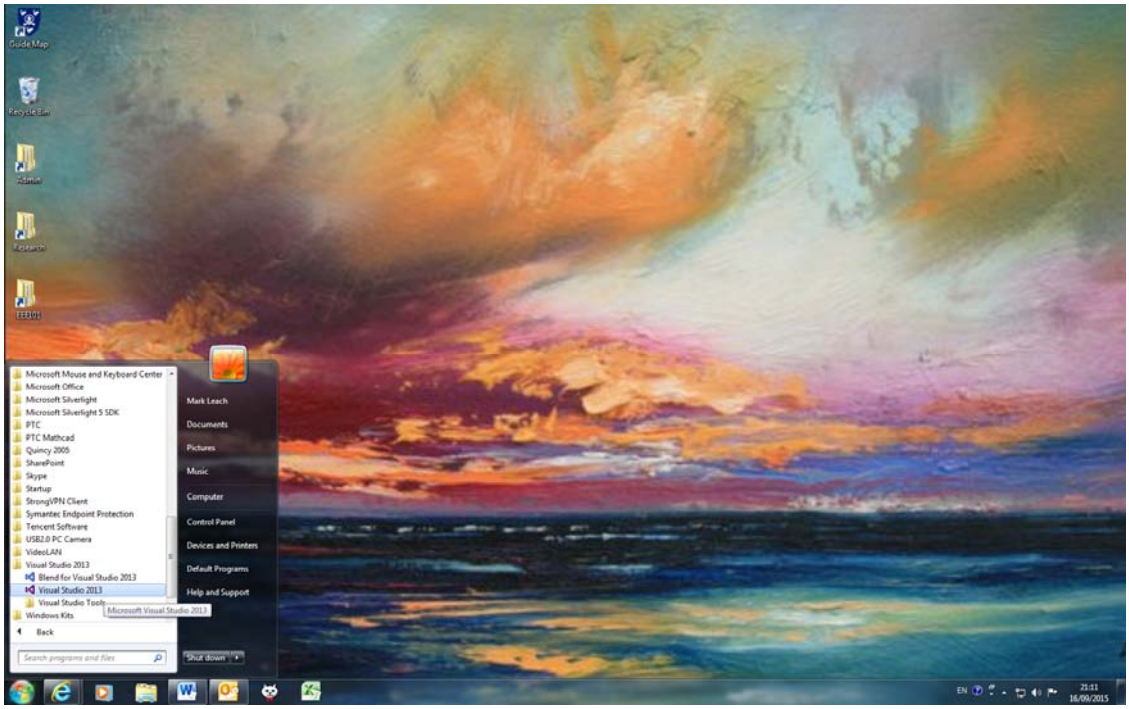
Next try to enter, compile and execute the C program shown below. Pay attention to the different sections of the program. Do not forget to type any symbols.


/*
* Converts distances from miles to kilometers.
*/
standard header file
comment
preprocessor directive
#include <stdio.h>
/* printf, scanf definitions */
#define KMS_PER_MILE 1.609
/* conversion constant */
constant
reserved word
int
main(void)
{
double miles,
/* distance in miles */
variable
kms;
/* equivalent distance in kilometers */
comment
/* Get the distance in miles. */
standard identifier
printf("Enter the distance in miles> ");
scanf("%lf", &miles);
/* Convert the distance to kilometers. */
kms = KMS_PER_MILE * miles;
special symbol
/* Display the distance in kilometers. */
printf("That equals %f kilometers.\n", kms);
reserved word
return (0);
punctuation
}
special symbol

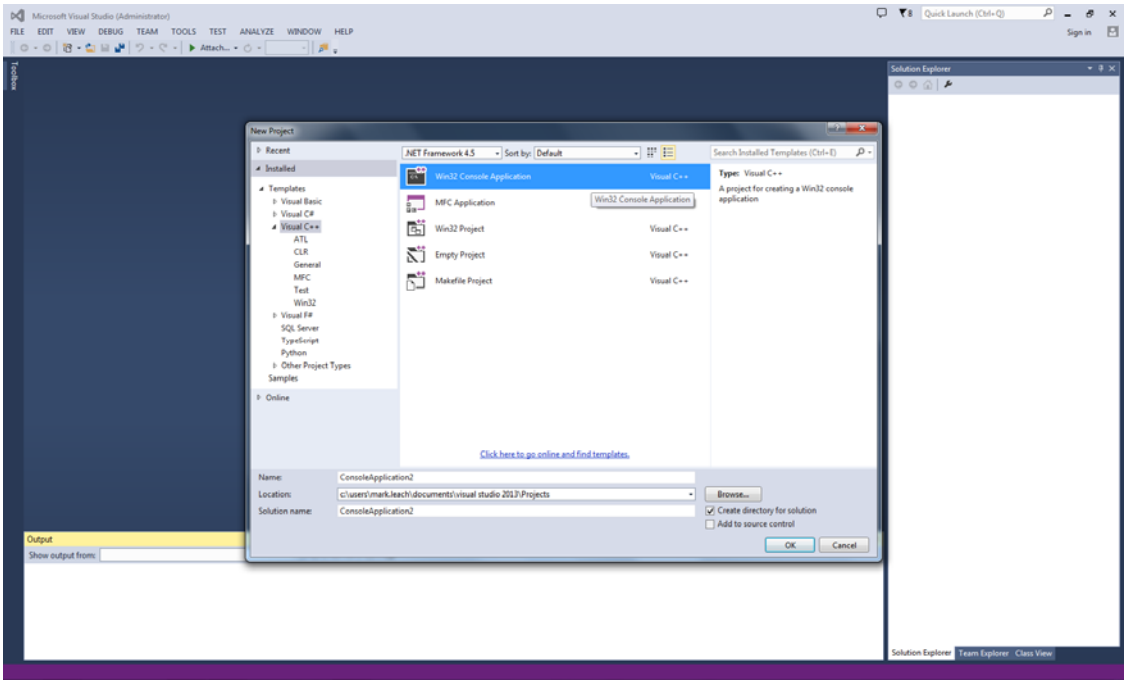
Using Visual Studio 2013 Full IDE

Now you should be familiar with the process of writing, compiling and running a simple program from the command line. You can now experience the same process using the full development environment available in Visual Studio 2013.

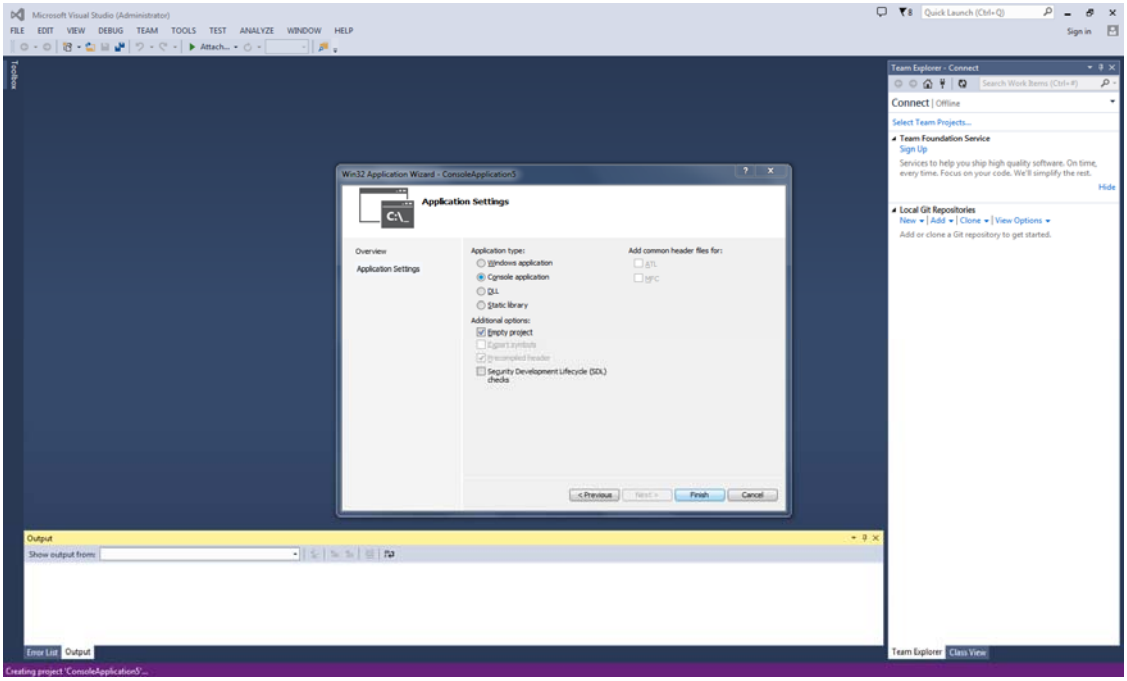
To open visual studio follow start>all programs>visual studio 2013> visual studio 2013 (See below)



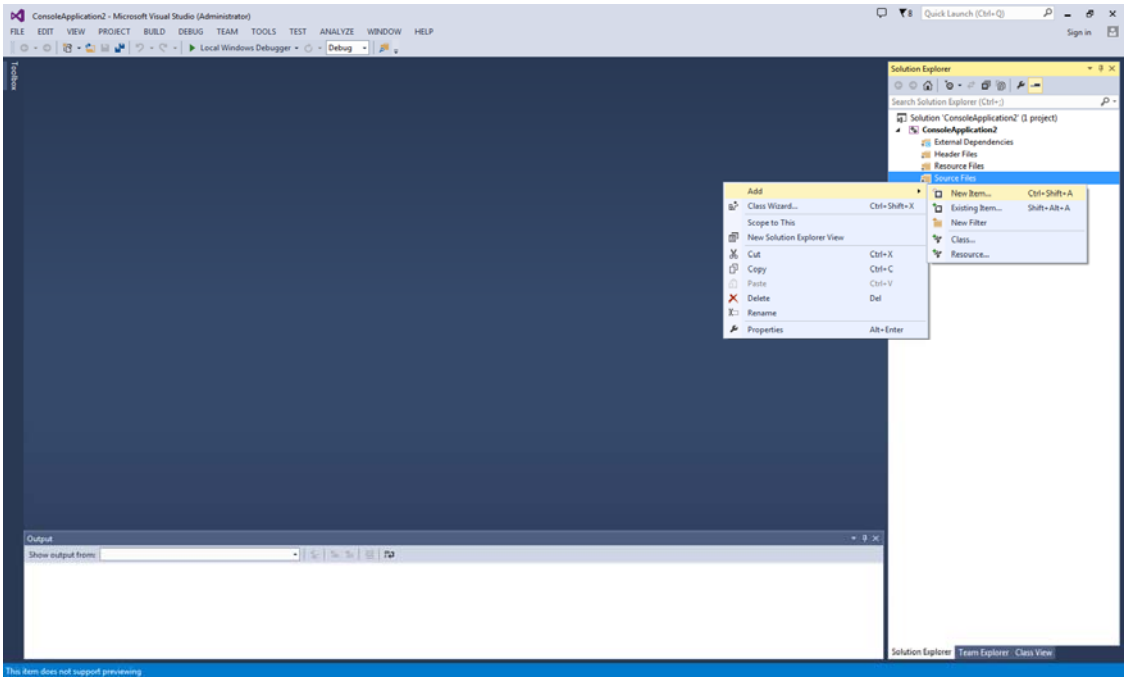
Once the application is open, create a new project (The first time you use the application it will provide you with a dialogue box to do this). On the drop down menu use file>new>project, you will see the following screen



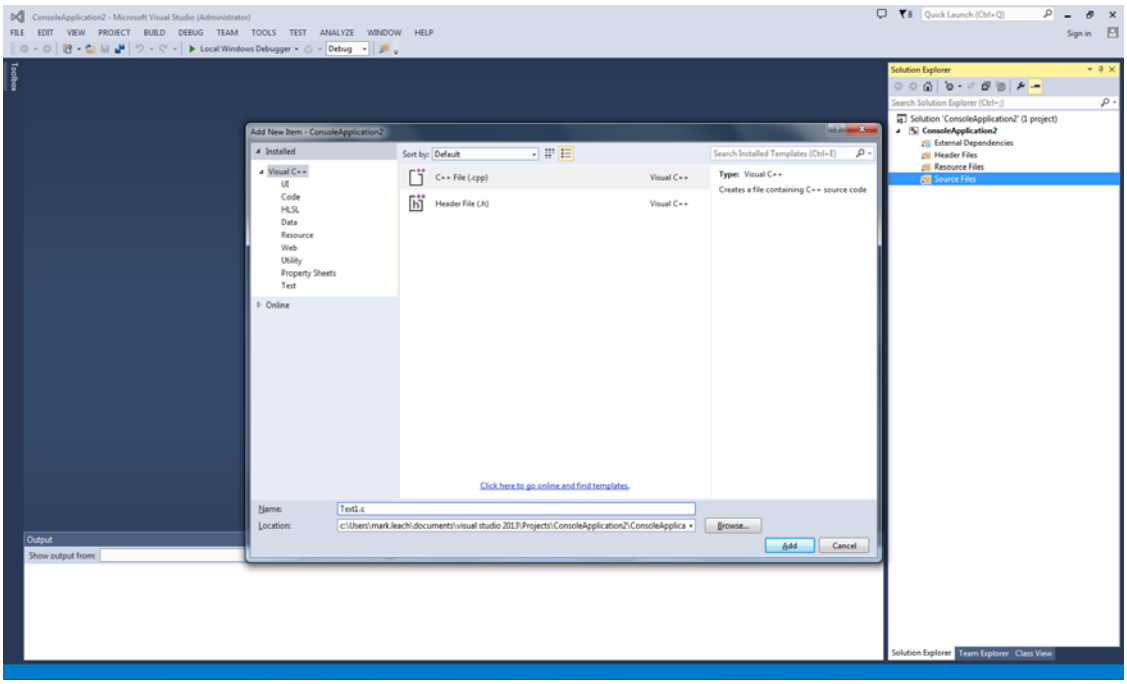
Select Visual C++ on the left hand menu and then Win32 Console Application and click ok. On the next dialogue screen select **Next**, and then on the following dialogue screen check the box that says **Empty Project** and **Deselect** **Security Development Lifecycle (SDL) checks** before clicking **Finish** (See below)



In the Solution Explorer window on the right hand side, the project files appear. The only one you need to worry about at this point is the source files. Right click on this and select Add>new item (See below)

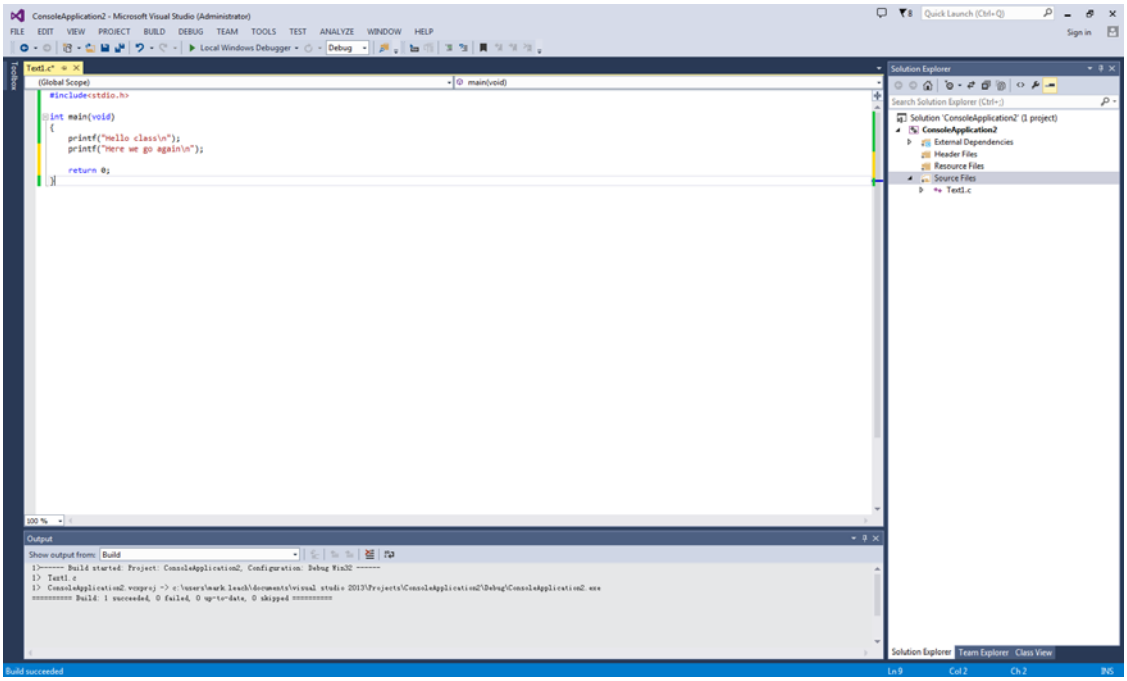


In the dialogue box that opens, select the C++ File and give your program a Name at the bottom of the window. Remember that you are using C not C++ so name your file filename.c (not .cpp) and press Add (See below).



You now have an editor window where you can type your source code. Note to see the output screen at the bottom, you can open it from View>output.

Once you have typed your code (See below) You can build it (same as compile) using the build menu, selecting Build>Build Solution or pressing ctrl+shift+B. The output box will show you if your build was successful or not and list any compile errors you have.



To run the program select the menu Debug>Start without debugging or press Ctrl+F5. If you have edited your program it will ask if you want to build it first.

You can build and run the program at the same time by selecting the above Ctrl+F5.

Repeat the two programs you created earlier for command line compile and execution using the Visual Studio IDE.

Other Compilers

If you are looking for a more simple Windows based compiler to use, you can try the Quincy 2005 software that is available on ICE together with its user instructions. Once again, you can use any compiler you prefer. If you program using only standard C coding, then your code will run on any C compiler.