

A Crash Course in Microsoft Visual C++ 2008

This document presents the most essential things to know in order to program with VC++. This document is intended to use with the provided “OpenGLTutorials.zip-file” and is part of the TDA 361 - Computer Graphics course at Chalmers.

Copying and modification of this material is free.

Download the OpenGLTutorials.zip file from the course home page:

CHALMERS
Computer Engineering
Computer Science and Engineering – Chalmers University of Technology and Göteborg University

TDA361 - Computer graphics 2009 lp1

Anmälningsskod (for www.studera.nu): CTH-23630
Examiner: [Ulf Assarsson](mailto:Ulf.Assarsson@chalmers.se)
uffe@chalmers.se

[Home](#) | [Schedule](#) | [Literature](#) | [Tutorials](#) | [Exam](#)

SCHEDULE:

- [Link to always up-to-date schedule.](#)
- All lectures are in ED, located in the EDIT-building at Campus Johanneberg
- [MAP](#) for lecture hall and tutorial rooms

Lectures
Some Mondays (will be announced later), 13-15 in HA1/HB1 - see schedule above
Wednesdays, 13-15 in HC4
Fridays, 13-15 in HC4
Tutorials with course assistants available
Mondays 15-17
Wednesdays 17-19 (15-17 people with laptops can present at our offices: 4114 + 4124.
Fridays 15-17
The following plan may change during the course. The links for the Bonus-OH are located under the table.
Bonus material is simply non-compulsory additional material that is fun or highlighting for the interested reader.
The column "Tutorial" states when it is wise to start working on the corresponding tutorial.
Lösenordsskyddade bonusfiler packas upp med lösenord "datagrafik".
All self-studies below are **non-compulsory**

NOTE: If you are using the 2nd edition of Real-Time Rendering, for the compulsory RTR-chapter hints, see last year's schedule [here](#).

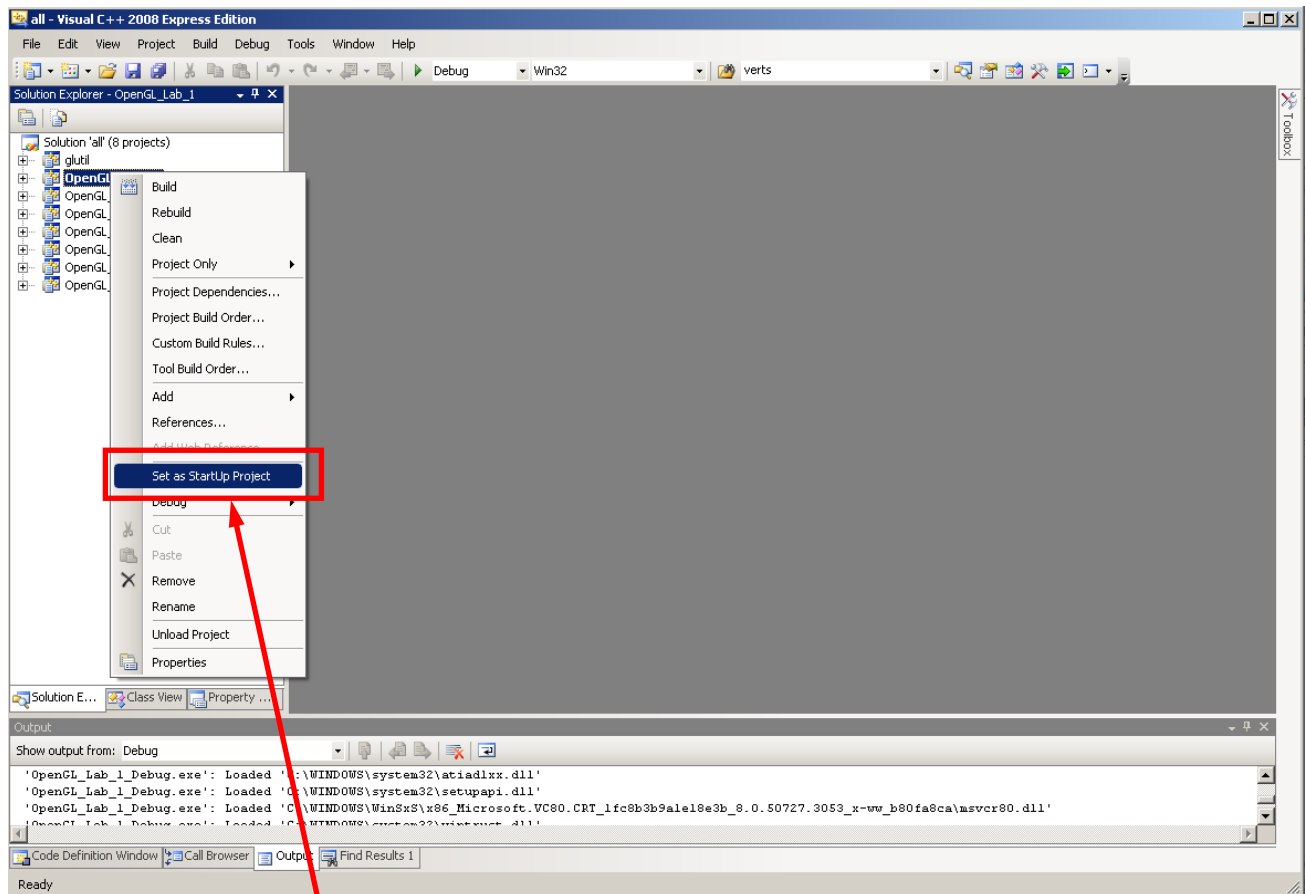
Lecture	Readings/Läsanvisningar	Tutorial	Deadlines
week 1			
Wed. lecture 1 - Introduction + Pipeline and OpenGL	RTR chapter 2, ch 15.2. pipeline.pdf Bonus: OH 1-16. simpleapp.zip - the test application shown at lecture, crash course in VC++ .zip. Also, see C++ introduction to C++ with example code .		
Self studies - Languages (non-compulsory)	Languages.pdf (in Swedish) - Read briefly in and only if you find it interesting		
Fri. lecture 2 - Vectors and Transforms	RTR ch 4: 4.1, 4.2.1, 4.2.4, 4.3 (briefly), 4.6, appendix A and B, vectors and transforms.pdf , Bonus: OH 27-44, "Från värld till skärm" förutom kapitel 10, "Introduktion till OpenGL" kap 1-9. Quick repetition of vector algebra: vectors.zip (lösenordsskyddad - se ovan).		

Bonus material:
OH1-16 OH17-48 OH49-64 OH65-88 OH89-104
OH105-128 OH129-144 OH145-176 OH177-184
OH185-200 OH201-224 OH225-248 OH249-256

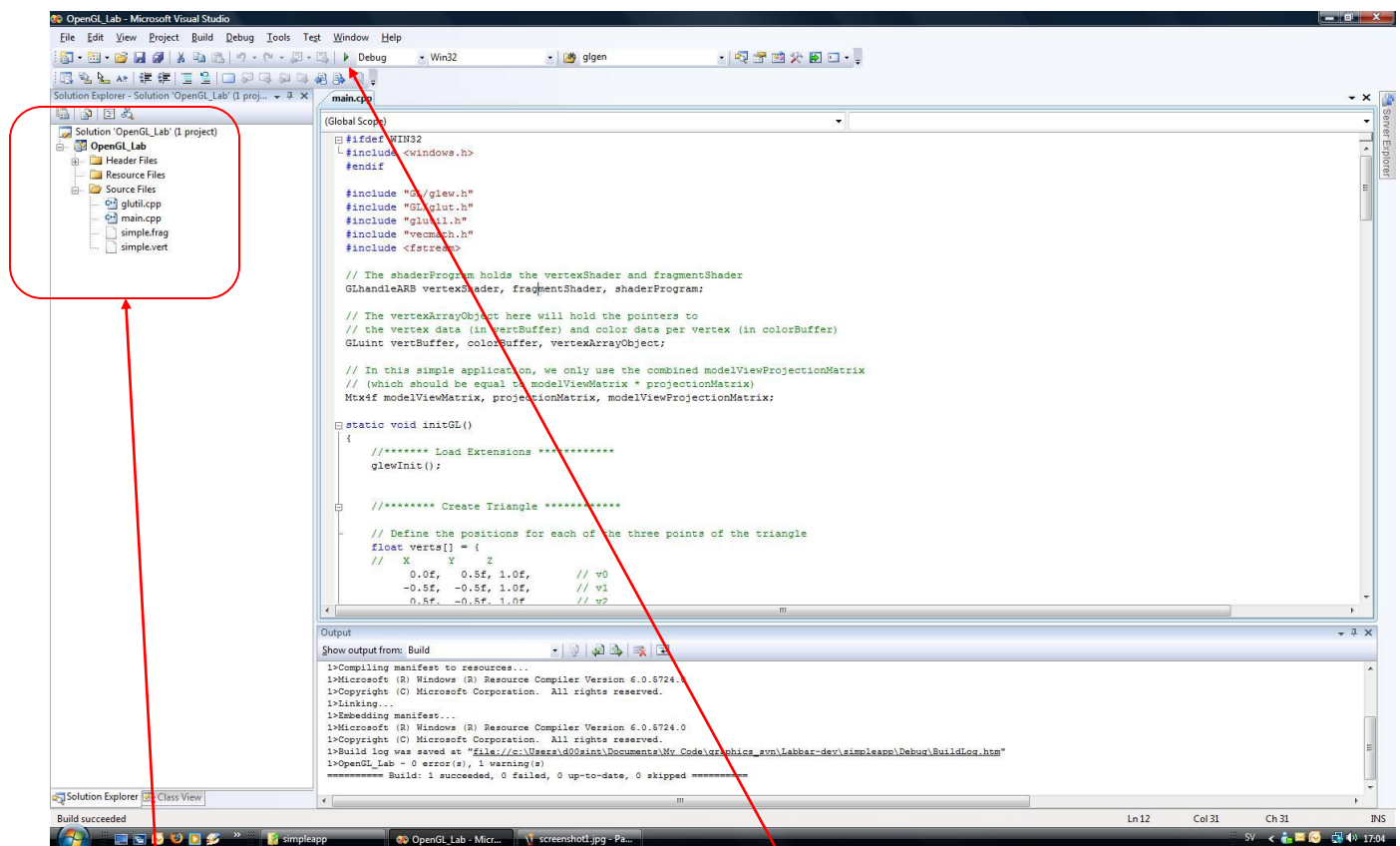
Unzip the OpenGLTutorials.zip -file and find the all.sln-file which is located in the "OpenGLTutorials"- directory where you chose to unzip. Simply double click this .sln file to start VC++. Should VC++ want to convert the project from 2005 to 2008 then let it do so.

NOTE: If this is the first time you start Visual Studio, be sure to select C++ developer profile (or whatever it is called).

After a few seconds a window similar to this should appear:



This solution contains all the OpenGL tutorials for the course. To select which of the project is to be active, right click on the project (in this case select OpenGL_Lab_1) and select “Set as StartUp Project”.

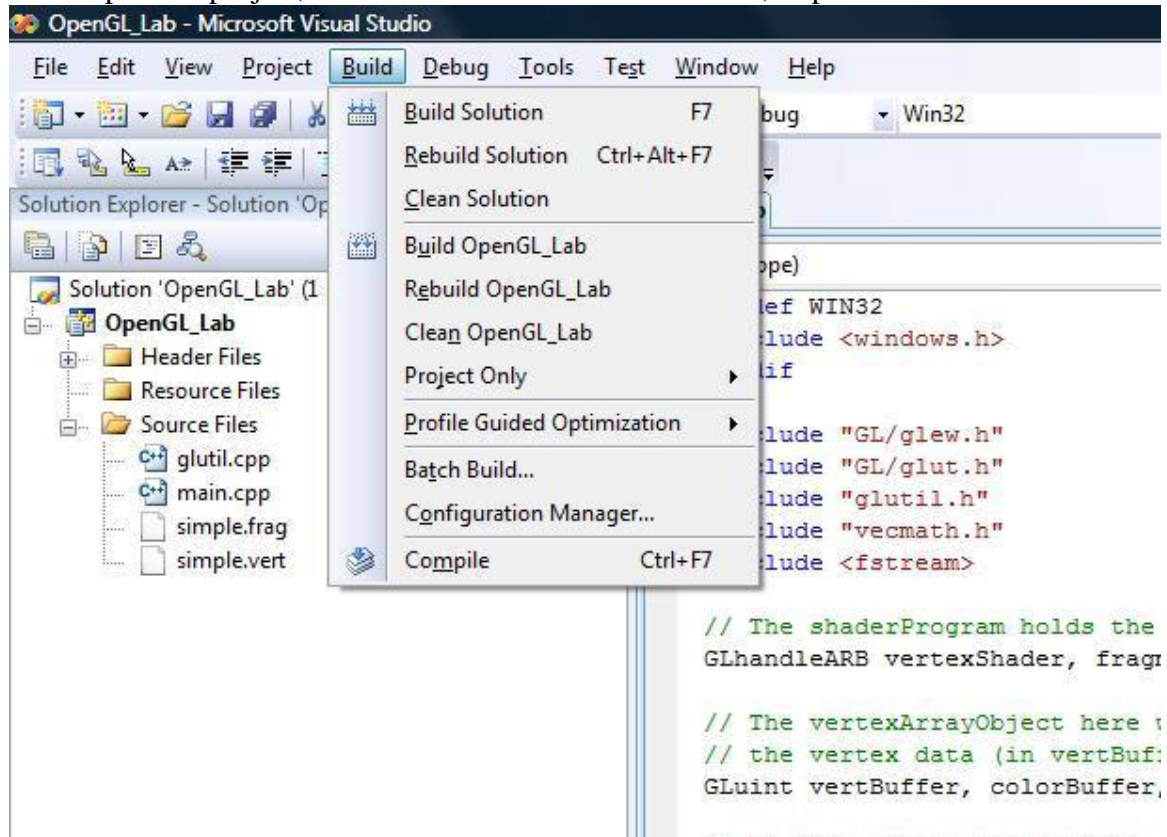


To the left, you see the files in the projects.

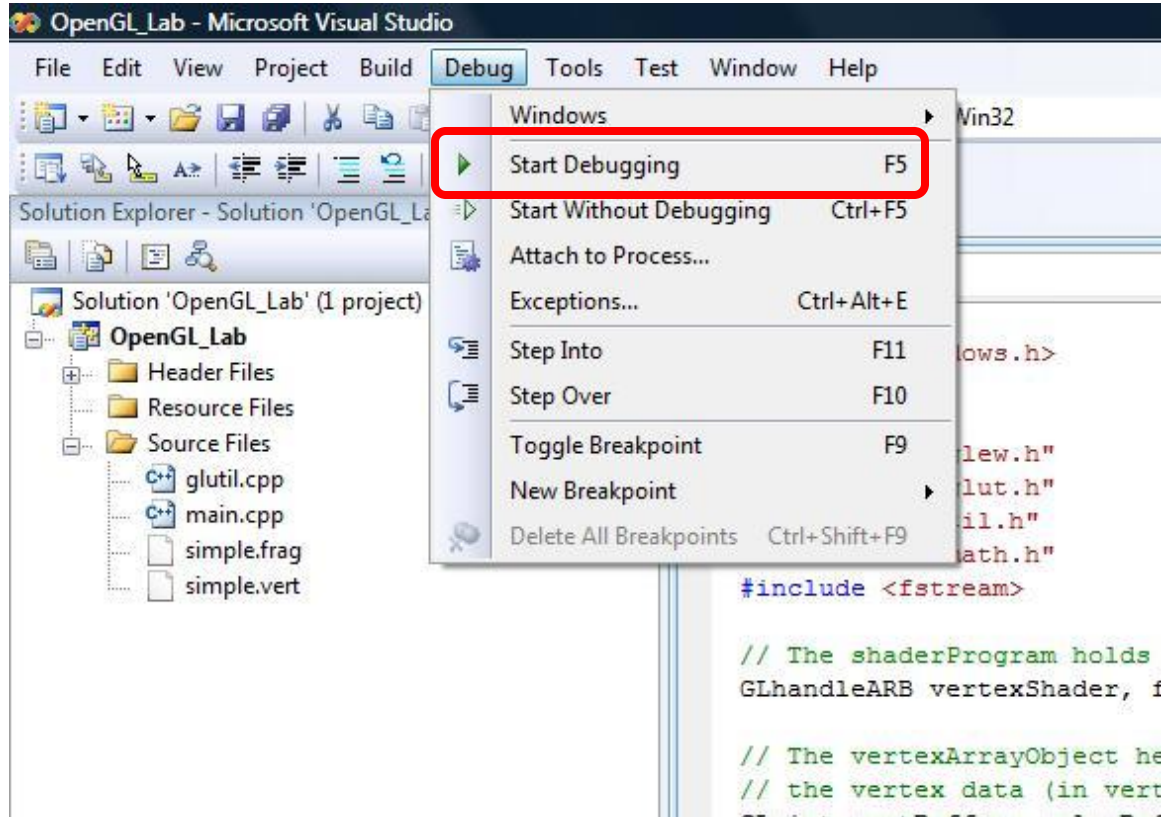
Compile and run the project by pressing F5, or clicking this “play” button with a green arrow.

When the program starts, you should see a white triangle on a blue background.

To compile the project, select menu Build->Build Solution, or press F7:



To run the application, press F5 or select menu Debug->Start.



The images below are from using VC++ 2003, but the principles are still the same.

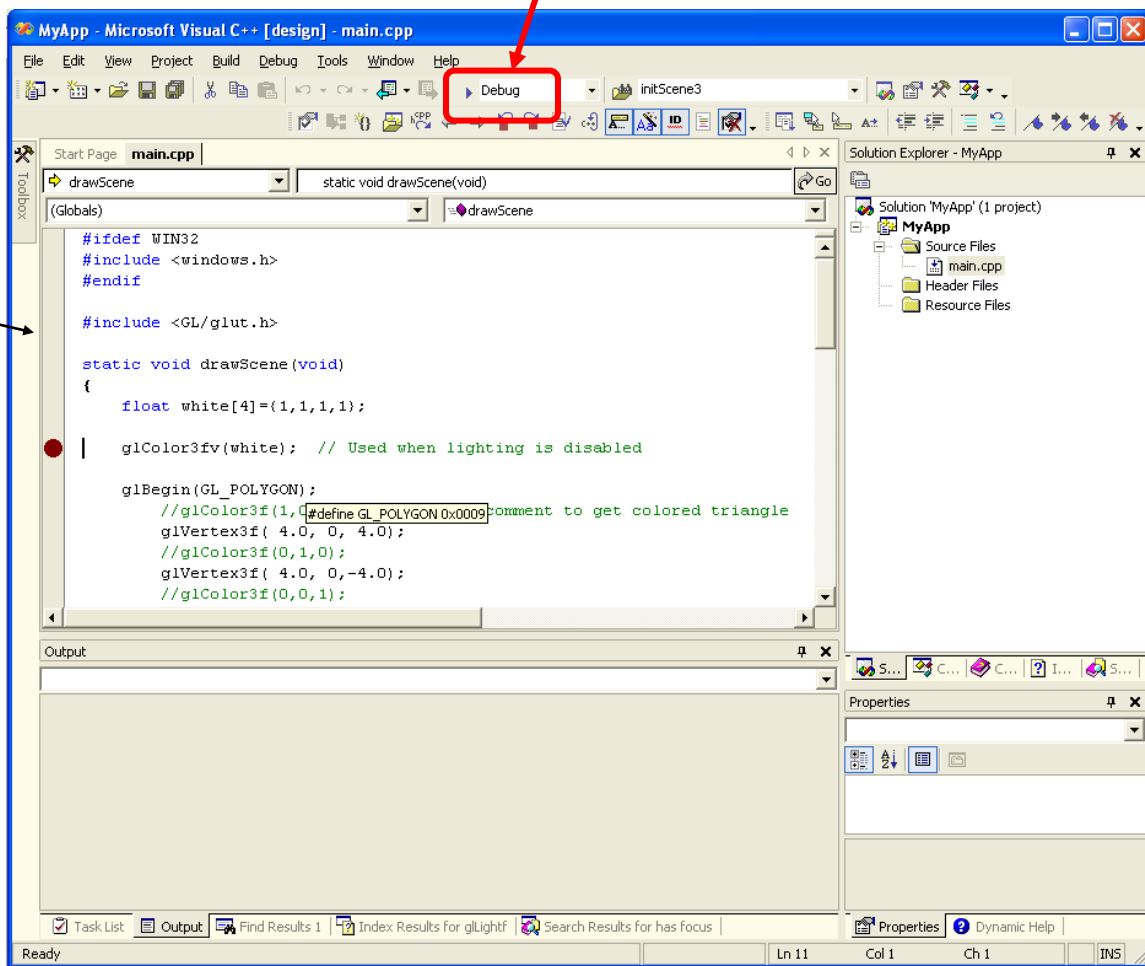
Using the debugger

Using the debugger is very easy and is highly recommended.

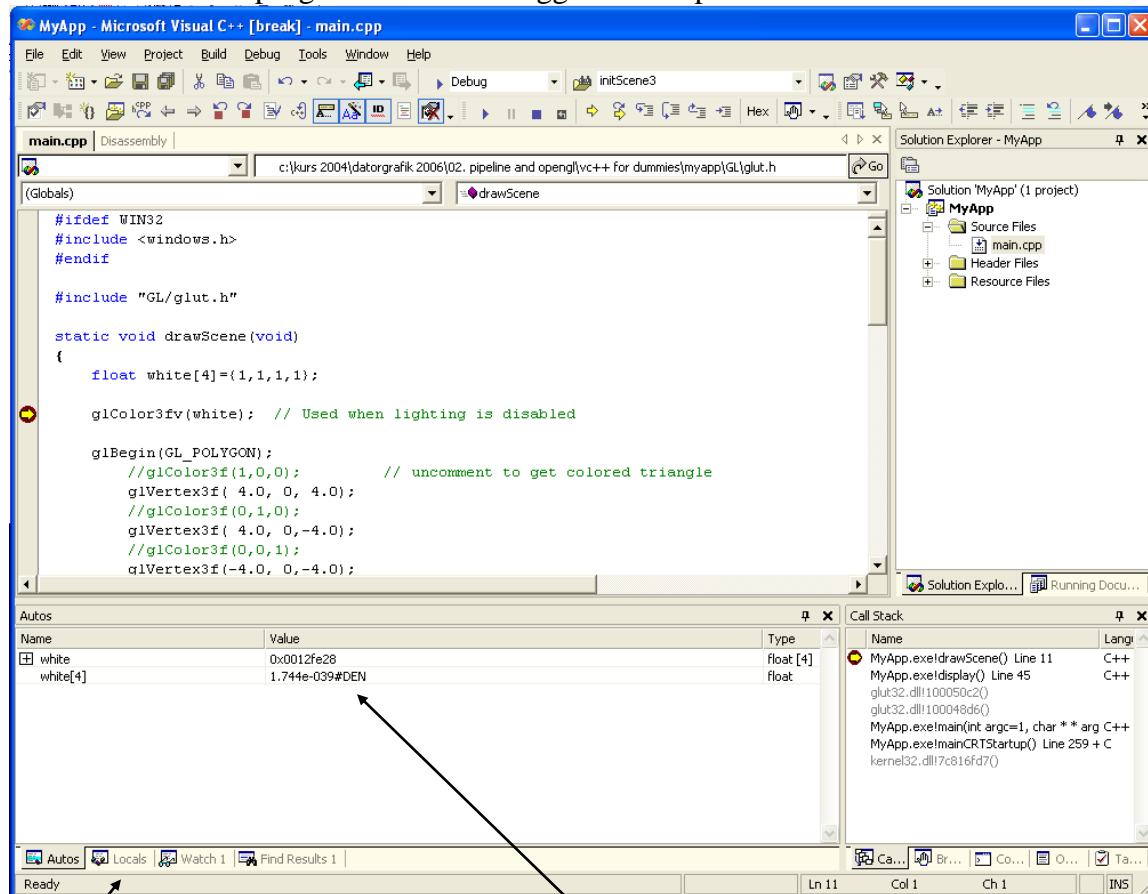
F9 sets/removes a break point.

Set a break point by pressing F9 at the desired location in your source code. Remove the breakpoint by pressing F9 at the same location.

Note: Ensure the project is set to build in debug mode; otherwise you will not be able to set a breakpoint or inspect variables.



Press F5 to run the program and the debugger will stop at the location. Do this!



The values of the most relevant variables are shown at tab “Autos” here. Note that the values can be modified directly by typing new values.

The values of all variables are shown at tab “Autos” here:

Summary of most common functions:

- F9 – sets/removes a break point at the location of the cursor
- F10 – steps one line in the source code
- F5 – continues the running of the program
- Shift + F5 – stops debugging
- F11 – steps into a function call if a function call is made at that line and source code for the function is available
- Shift + F11 – steps out from a function call

Miscellaneous about C++

Typically you store the class definition in a `class_name.h` file and the implementation of the methods (functions) in a `class_name.cpp` file.

The class definition in the `.h`-file looks like this:

```
// The #ifndef avoids the problems with multiple includes of the
// same file. Otherwise, the compiler may complain that the class
// is already defined.
#ifndef YOUR_CLASS_NAME_AND_SOMETHING_H
#define YOUR_CLASS_NAME_AND_SOMETHING_H

class Example
{
public:
    // Regarding speed it is good to inline the
    // constructors if they are short. Here they are empty.
    Example() {}

    // And the same applies for destructors
    ~Example() {}

    bool Method1();

private:
    int m_a;
};
#endif //YOUR_CLASS_NAME_AND_SOMETHING_H
```

The `class_name.cpp` file looks like this:

```
#include "class_name.h"
#include <stdio.h>

bool Example::Method1()
{
    printf("Hi");
    return true;
}
```