MA400: Financial Mathematics

Introductory Course

Lecture 1: Overview of the course

Preliminaries

A brief introduction

Beginning to program

Some example programs

Aims of this course

Students should have a familiarity with the basics of C++ programming. In particular, you will see

- fundamental data types
- arithmetic and operators
- tests and loops
- functions (encapsulating code)
- pointers and arrays

That is, you will see the C of C++.

However, this is not a software programming course.

Recommended reading

Find one that suits your learning requirements.

- Introducing C++ for Scientists, Engineers and Mathematicians, 2nd Ed. D.M.Capper.
- The C++ Programming Language, Special Edition. Bjarne Stroustrup.
- Online C++ language tutorial at http://www.cplusplus.com/doc/tutorial/
- Thinking in C++ Vols 1 & 2. Bruce Eckel. http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html

There are also a lot of free online tutorials and resources.

Structure of this course

This is a 11 hour course, of which

- 2 hour (supervised) lab session [worksheet]
- 9 hours of lectures
  - Week 1 (3 hours):
    - Introduction to the course
    - Fundamental types and operators
    - Control structure
  - Week 2 (6 hours):
    - Functions
    - Pointers and arrays
    - Summary and tips. A look forwards.
- MA417 - Computational Methods in Finance

What is C++?

C++ is a general purpose programming language that is itself a superset of the C programming language.

It is a compiled language, so that source files are used to generate executables, rather than executed themselves.

It is a general-purpose programming language that is designed to support:

- data abstraction
- object-oriented programming,
- generic programming.

Data abstraction

User-defined *types* which allow the user to model a particular entity in their system, together with various operations on it.

This is done through **classes**

For example:

- matrices,
- complex numbers,
- even the term structure of interest rates!

The term structure of interest rates as an object

Note that the term structure of interest rates can be equivalently specified in terms of either

- discount factors,
- spot interest rates, or
- forward interest rates.

Complex numbers as an object

```
class complex {

private:
  double re, im;

public:

  complex(double r, double i) { re = r ; im = i ; }
  complex() { re = 0 ; im = 0 ; }

  friend complex operator+(complex, complex);
  ...

  friend bool operator==(complex, complex);

  ...
};
```

Complex numbers as an object

```
complex operator+(complex a1, complex a2)
{
  return complex ( a1.re + a2.re, a1.im + a2.im );
}
```

Object-oriented programming

Consider the following situations

- Suppose we have defined a Shape class, and alongside this we also had a Disc class, a Triangle class and a Square class.
- Consider a Matrix class, and a class representing all invertible matrices.
- Consider a Vector class, and a class representing all 3-vectors.

In all these cases, the latter classes clearly inherit, or derive, properties from the former, parent, class.

Languages which allow such class hierarchies to be expressed and used support **object-oriented programming**.

Generic programming

Using **templates**, or **parameterised types**, C++ allows us to implement algorithms that are independent of the data types they are used on.

E.g. arrays, lists and vectors are all data structures which are amenable to sorting, copying and searching functions.

Integers and floating numbers are also amenable to the same sorts of functions and operators, such as finding the maximum of two numbers.

Programming experience

- C/C++
- Java
- Other (Perl, Python, etc)
- Maple/Mathematica/Matlab/etc

Learning to program

Program.

Think.

Document.

Your programming environment

What do you need to start programming in C++?

- A pencil and some paper.
- A text-editor to write the source code
- A compiler (GCC)
- Access to the command prompt

However, you may prefer to work within an IDE (Integrated Development Environment)

- Bloodshed Dev-C++
- Eclipse

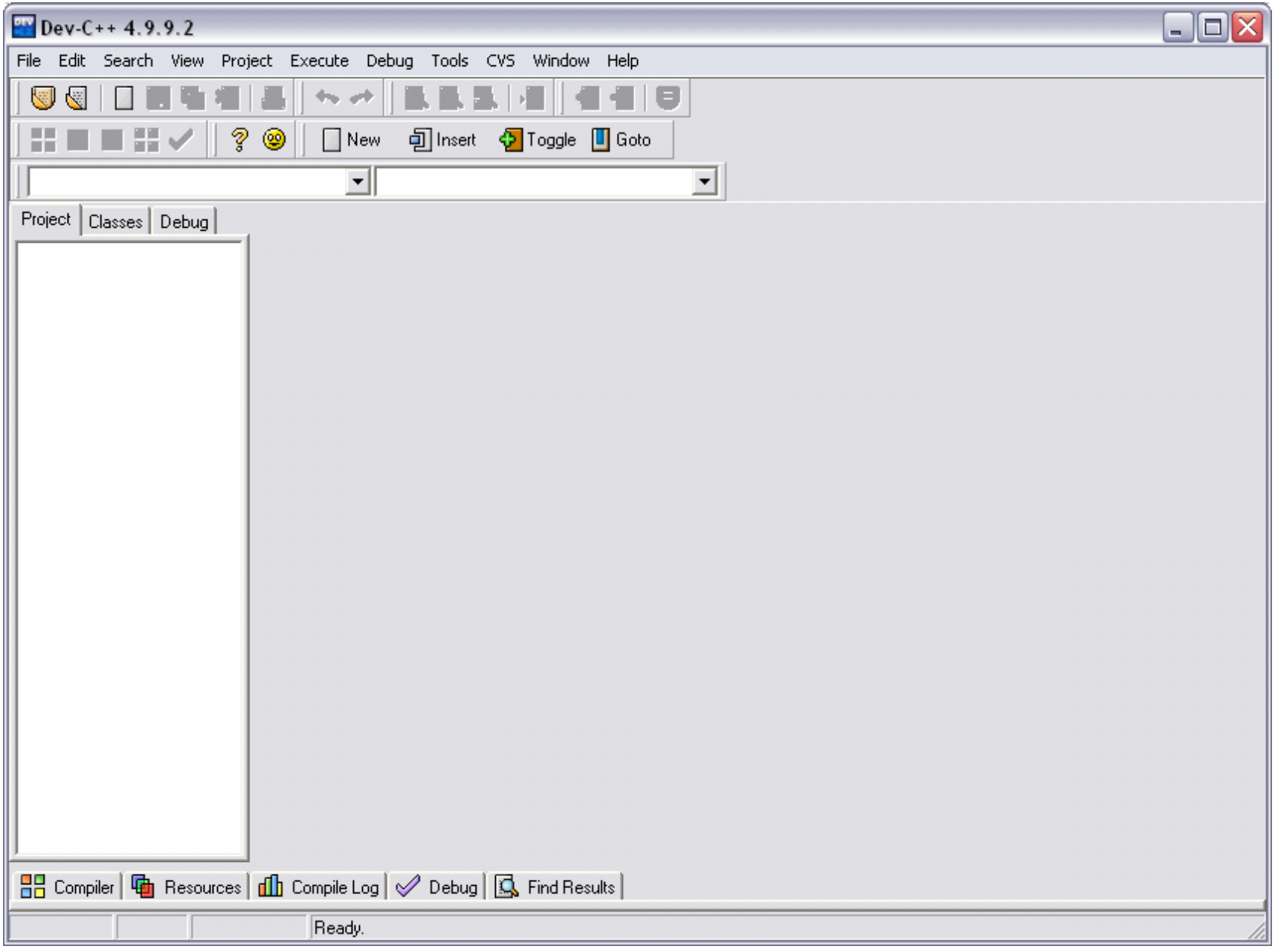
A comparison of various C/C++ IDEs can be found at Wikipedia: http://tinyurl.com/comparison-of-cpp-ides

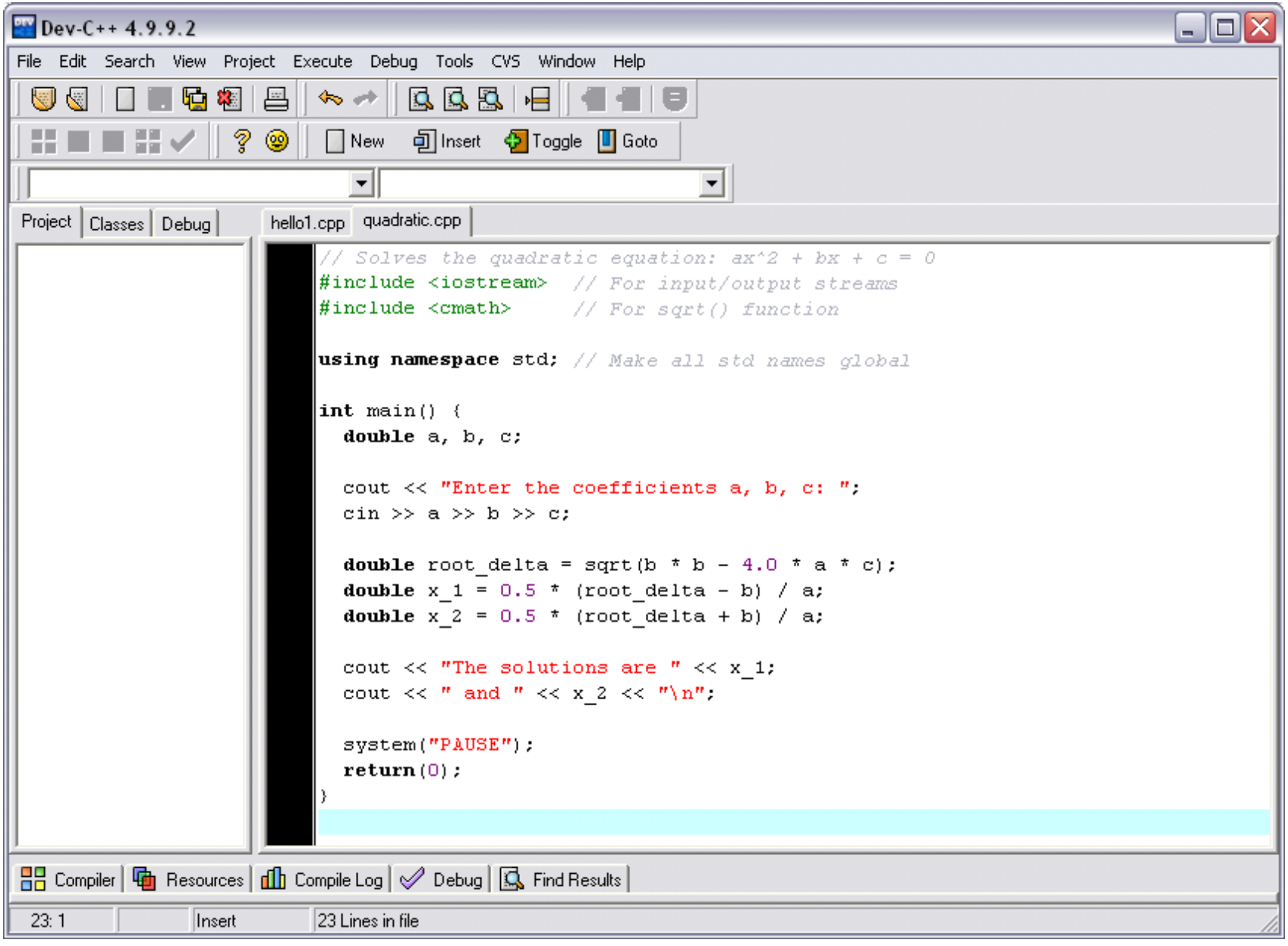
The simplest program possible

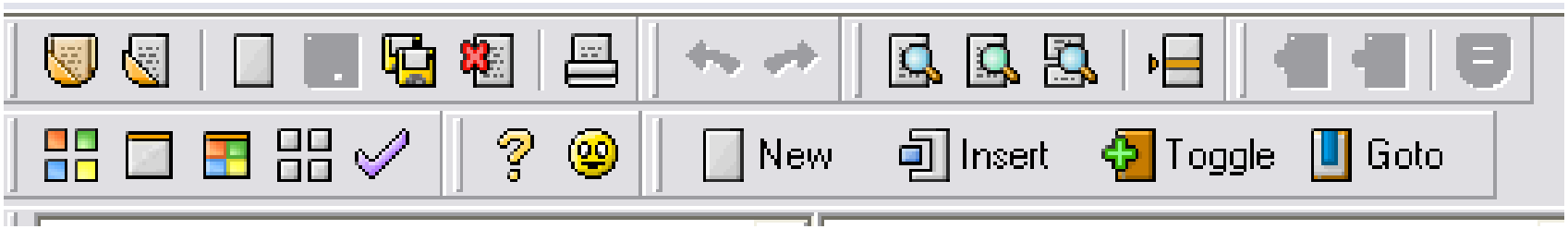
```
int main() {}
```

Hello, world!

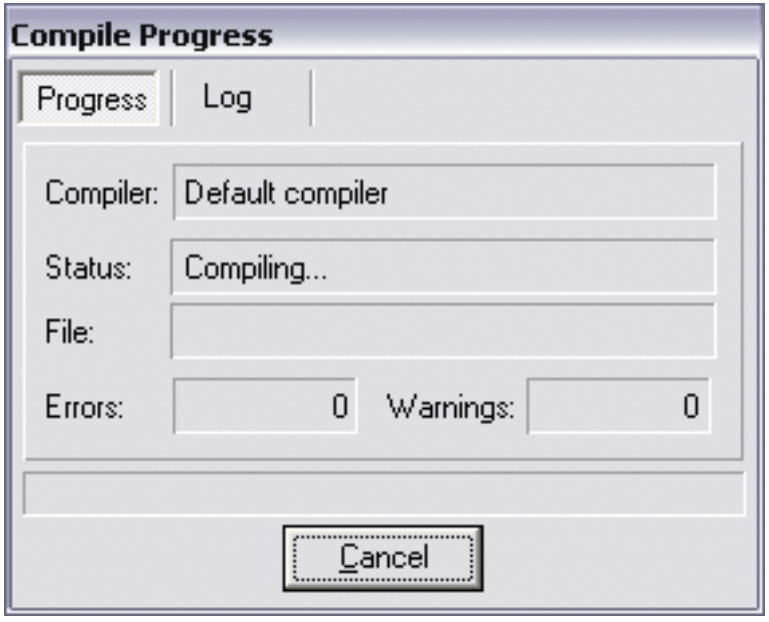
```
// Print "Hello, world!" to the standard output stream

#include <iostream>   // I/O stream facilities

int main()
{
  /*
     Use "put to" operator >> to print "Hello, world!"
     to standard output stream
  */
  std::cout << "Hello, world!\n";     // Note the newline

  // Return 0 to indicate successful execution
  return(0);
}
```

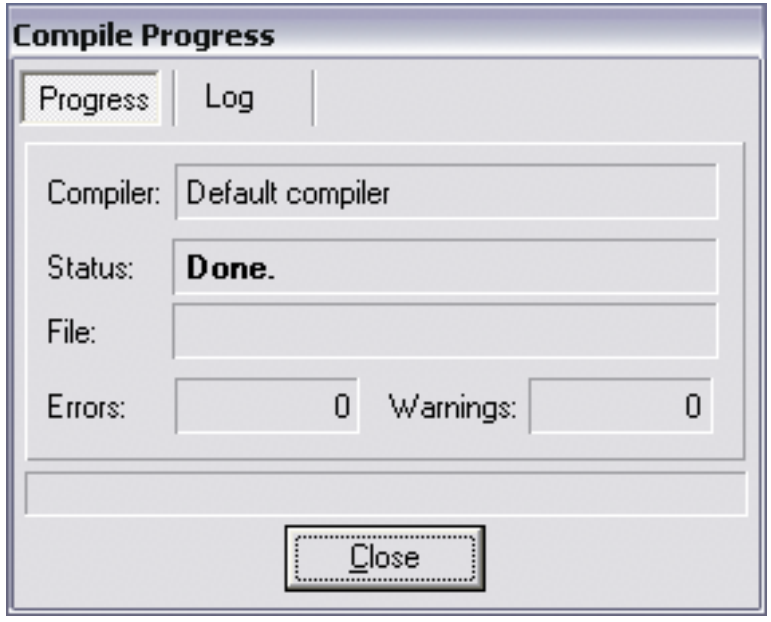

Dev-C++ 4.9.9.2
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
Compiler Resources Compile Log Debug Find Results
Ready.


Dev-C++ 4.9.9.2
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
hello1.cpp quadratic.cpp
// Solves the quadratic equation: ax^2 + bx + c = 0
#include <iostream> // For input/output streams
#include <cmath> // For sqrt() function
using namespace std; // Make all std names global
int main() {
double a, b, c;
cout << "Enter the coefficients a, b, c: ";
cin >> a >> b >> c;
double root_delta = sqrt(b * b - 4.0 * a * c);
double x_1 = 0.5 * (root_delta - b) / a;
double x_2 = 0.5 * (root_delta + b) / a;
cout << "The solutions are " << x_1;
cout << " and " << x_2 << "\n";
system("PAUSE");
return(0);
}
Compiler Resources Compile Log Debug Find Results
23: 1 Insert 23 Lines in file



- Compile (Ctrl + F9)
- Run (Ctrl + F10)
- Compile & Run (F9)


Compile Progress
Progress
Log
Compiler:
Default compiler
Status:
Compiling...
File:
Errors:
0
Warnings:
0
Cancel


Compile Progress
Progress
Log
Compiler:
Default compiler
Status:
Done.
File:
Errors:
0
Warnings:
0
Close

Running from the command line

```
H:\>dir
09/11/2009  05:24 PM    <DIR>          .
09/11/2009  05:24 PM    <DIR>          ..
09/11/2009  05:23 PM               576 hello.cpp
09/11/2009  05:24 PM           475,852 hello.exe
               2 File(s)        476,238 bytes
               2 Dir(s) 37,365,292,800 bytes free

H:\>quadratic.exe
```

Compiling from the command line

Using the Gnu C Compiler (gcc):

- Windows

  ```
  C:\Dev-Cpp\bin\g++ -o hello hello.cpp
  ```

- Linux, Mac OS X

  ```
  gcc -lstdc++ -o hello hello.cpp
  ```

A basic program template

```
// Describe the program's function
#include <iostream>     // For input/output streams
#include <cstdlib>      // For EXIT_SUCCESS
using namespace std;    // Make all std names global

int main()
{

  // Program code

  system("PAUSE");
  return(EXIT_SUCCESS);
}
```