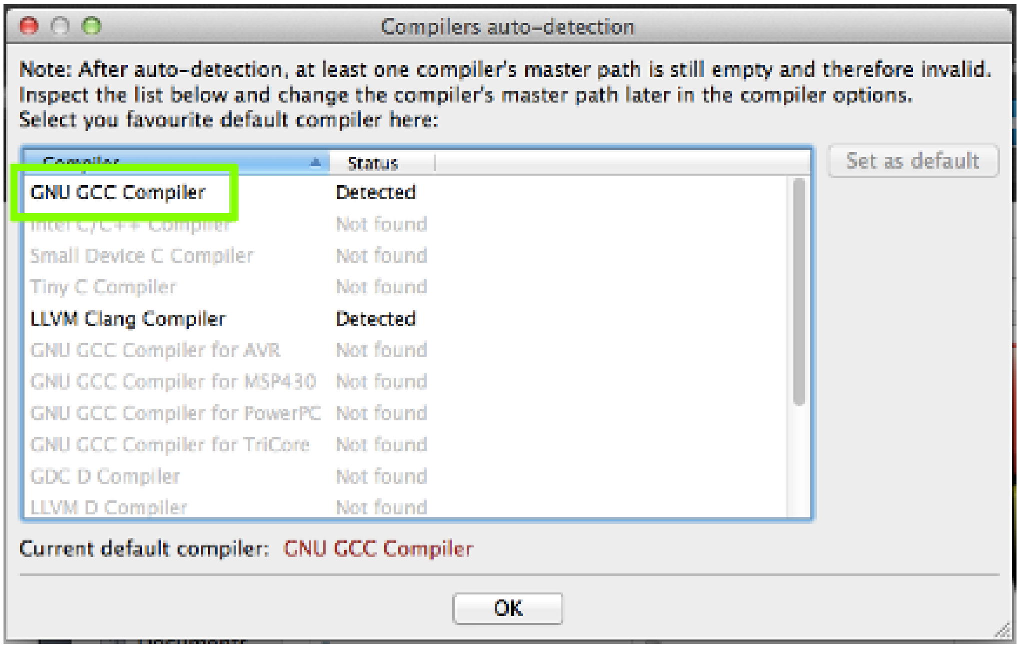
Project 1

due at midnight on Mon 8 Sep (60 points)

For this course, we will use the **Code::Blocks** development environment. It works on Windows, Mac, and Linux. You can download it from http://www.codeblocks.org/downloads/26:

- For Windows, select the one named codeblocks-VERSION-mingw-setup.exe, and run the installer.
- For Mac, there is a zip file that contains an Application bundle directly. When you run it, it may ask about the compiler to use:



Compilers auto-detection

Note: After auto-detection, at least one compiler's master path is still empty and therefore invalid.
Inspect the list below and change the compiler's master path later in the compiler options.
Select you favourite default compiler here:

Compiler	Status
GNU GCC Compiler	Detected
Intel C/C++ Compiler	Not found
Small Device C Compiler	Not found
Tiny C Compiler	Not found
LLVM Clang Compiler	Detected
GNU GCC Compiler for AVR	Not found
GNU GCC Compiler for MSP430	Not found
GNU GCC Compiler for PowerPC	Not found
GNU GCC Compiler for TriCore	Not found
GDC D Compiler	Not found
LLVM D Compiler	Not found

Current default compiler: GNU GCC Compiler

Figure 1: Choose GNU GCC compiler

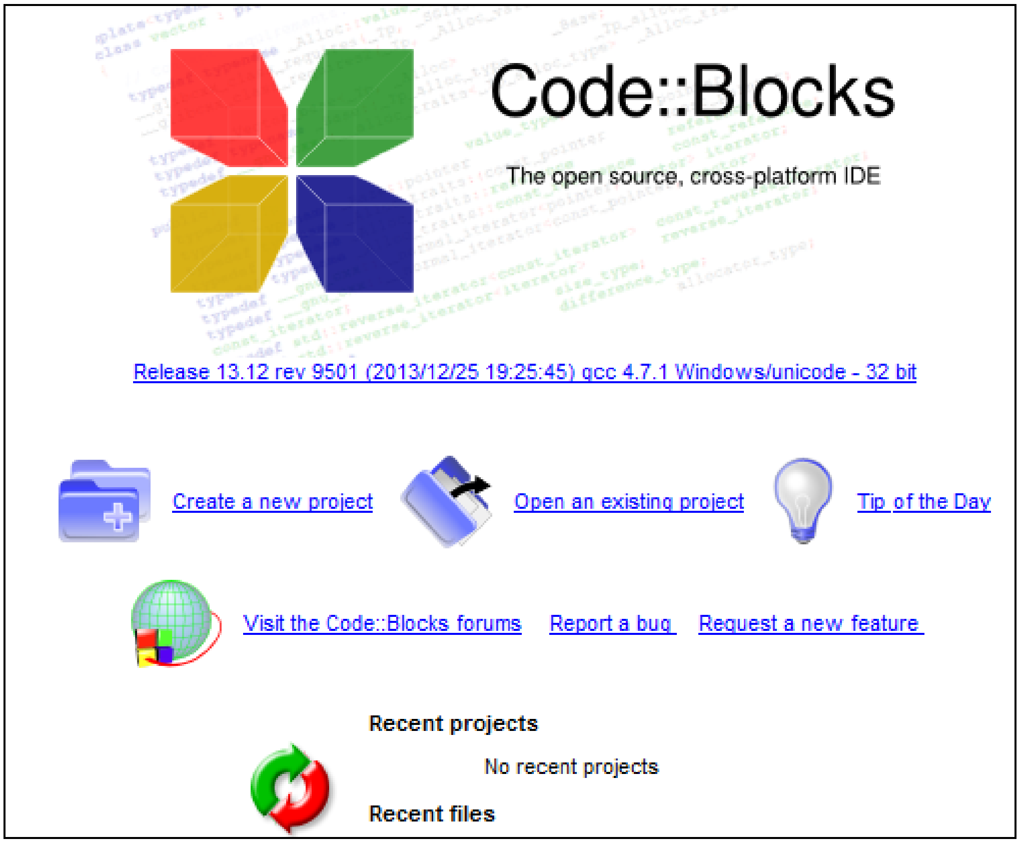
If it reports that no compilers are available, let me know right away.

New: if CodeBlocks is giving you trouble on the Mac, here are some instructions for using Xcode.

Get started with Code::Blocks

When you start Code::Blocks, you should see this welcome screen, with toolbars above it and a tabbed panel below.

Here is the simplest way to start programming with Code::Blocks:



Code::Blocks

The open source, cross-platform IDE

Release 13.12 rev 9501 (2013/12/25 19:25:45) gcc 4.7.1 Windows/unicode - 32 bit

Create a new project

Open an existing project

Tip of the Day

Visit the Code::Blocks forums

Report a bug

Request a new feature

Recent projects

No recent projects

Recent files

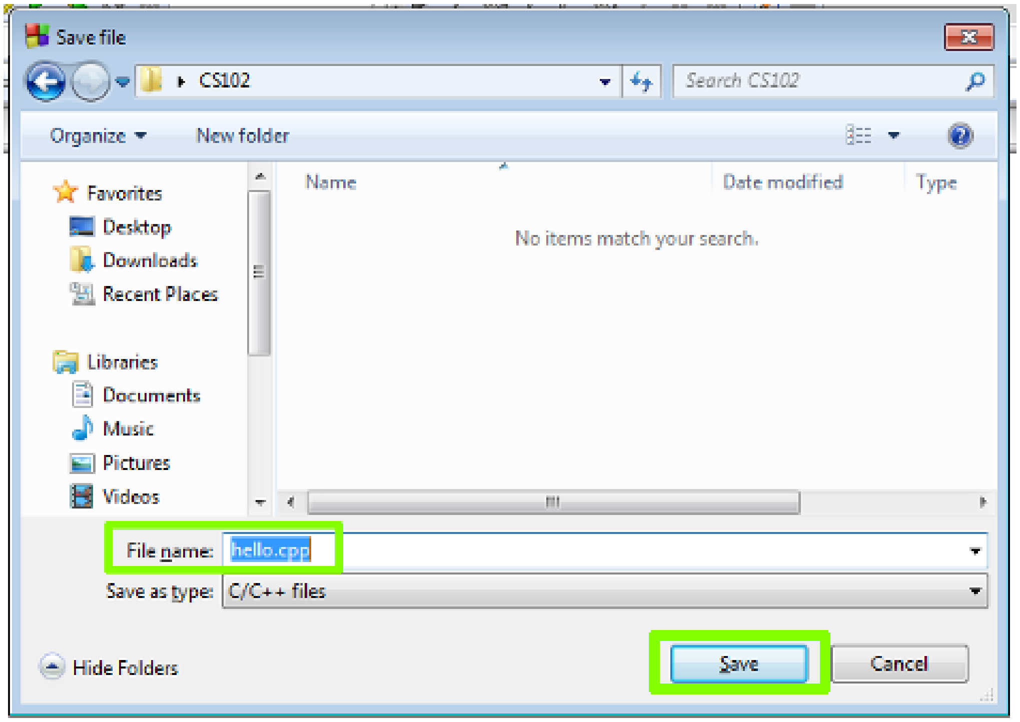
Figure 2: Code::Blocks Start Here

1. Select **File** » **New** » **Empty file...** from the menu (or hit Shift-Ctrl-N).

2. Select **File** » **Save file** from the menu (or hit Ctrl-S).

3. Navigate to someplace sensible (such as your Desktop or Documents folder), and create a new folder called **CS102**.

4. With that new folder selected, type the File name `hello.cpp` and hit Save.



File name: hello.cpp

Save as type: C/C++ files

Figure 3: Save hello.cpp

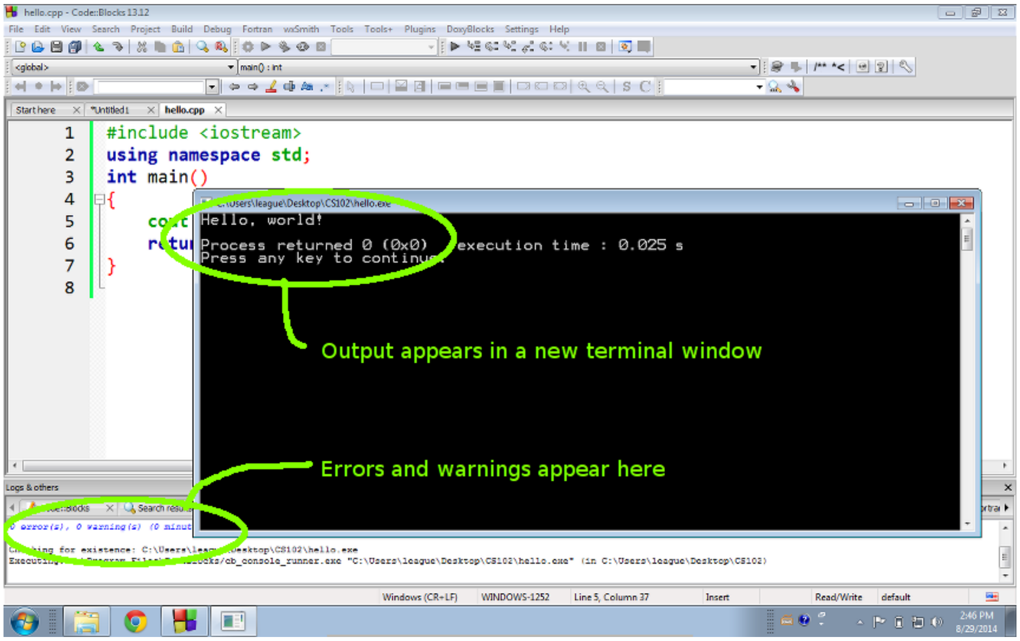
5. Now you can begin typing your first program into Code::Blocks. Below is an example you can use.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

As you type, the environment will offer possible completions of identifiers and keywords. You can use the arrow keys to select them, or just ignore the prompt. By default, it also inserts matching delimiters for you, such as entering both { and } when you just type {.

6. Save the file again, using Ctrl-S.

7. Now, select **Build** » **Build and run** from the menu, or press F9.



hello.cpp - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main() : int
Start here
*Untitled1
hello.cpp
C:\Users\league\Desktop\CS102\hello.exe
Logs & others
Search results
Windows (CR+LF)
WINDOWS-1252
Line 5, Column 37
Insert
Read/Write
default
2:46 PM
8/29/2014

```
#include <iostream>
using namespace std;
int main()
{
    cout
    retu
}
```

```
Hello, world!

Process returned 0 (0x0)   execution time : 0.025 s
Press any key to continue.
```

Output appears in a new terminal window

Errors and warnings appear here

```
0 error(s), 0 warning(s) (0 minut
Checking for existence: C:\Users\league\Desktop\CS102\hello.exe
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe "C:\Users\league\Desktop\CS102\hello.exe"  (in C:\Users\league\Desktop\CS102)
```

Figure 4: Output after build and run

8. Once your “hello world” program is working, you can move on to the next section.

Write a program to calculate age

Create a program that declares and initializes two integer variables:

- `birthYear`, which represents the year in which you were born
- `currentYear`, which represents the current year (2014)

The the program should do a calculation and print a message like this:

```
You turn 19 in 2014.
```

In this example output, we had `currentYear` set to 2014, and `birthYear` set to 1995.

Your program should adjust its output, however, depending on the values of those variables. For example, if I go into your program and change `currentYear` to 2254 and leave `birthYear` as 1995, it should say:

```
You turn 259 in 2254.
```

Submit your program

Please save your program as `p1age.cpp`, using all lower-case and no spaces. Then upload that file to Blackboard for project 1.