

# CSE 1061 Fundamentals of Programming

## Lecture #2

Spring 2015



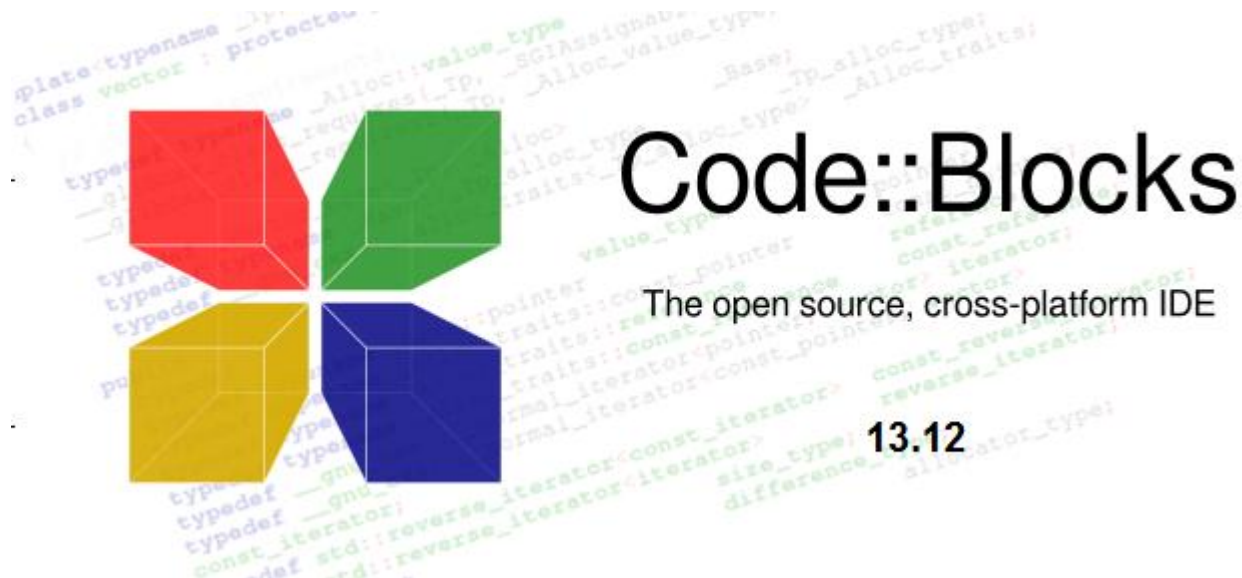
Computer Science & Engineering Program  
The School of EE & Computing  
Adama Science & Technology University

# Getting Familiar with the C++ Environment

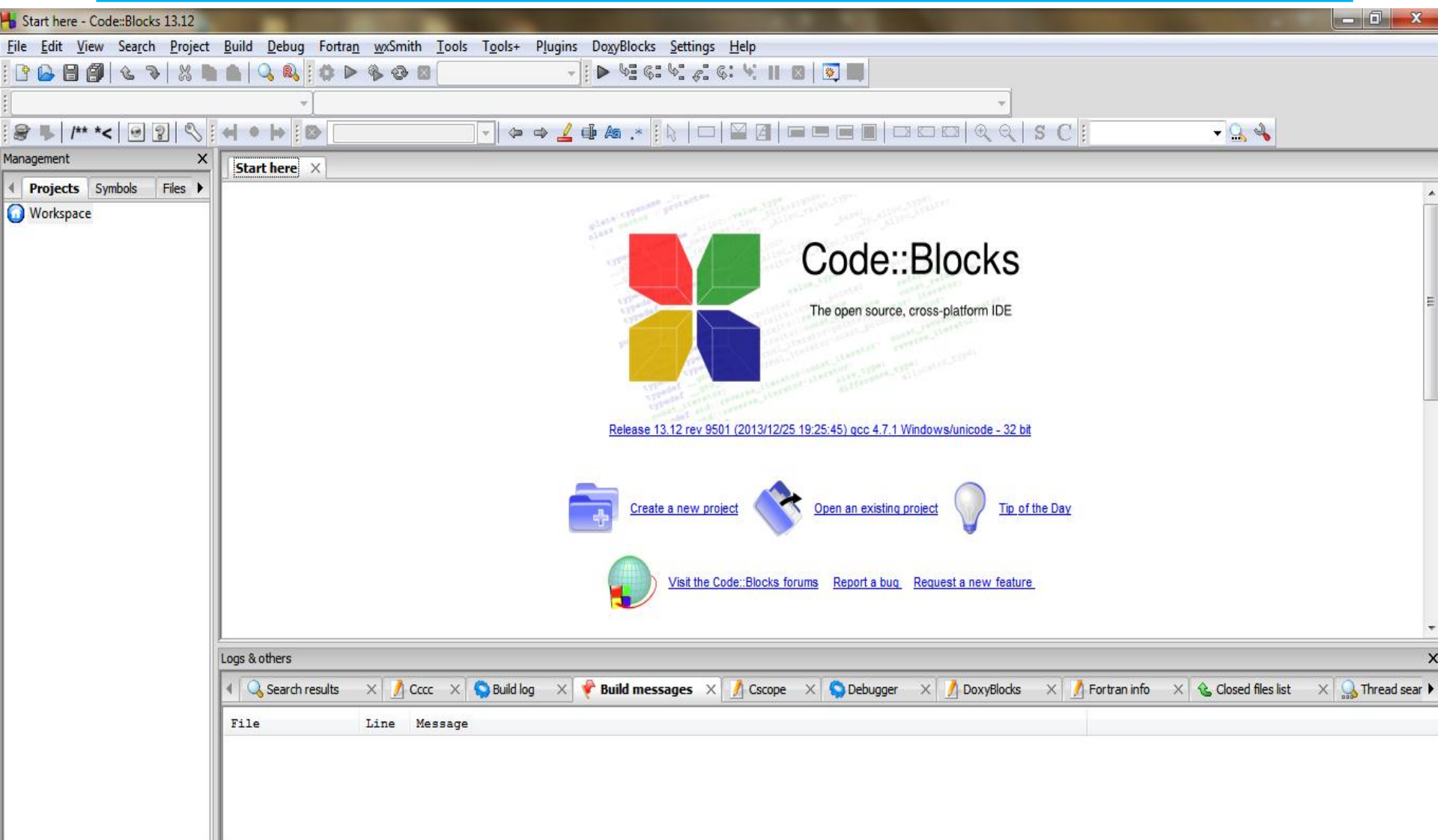
- Software Requirement
- Opening an IDE
- Creating new Project
- Workspace and Projects manager
- Changing Code Blocks Settings
- Our First C++ Program

- A C++ Compiler
  - **GCC(MinGW) Compiler<=Preferred**
  - Microsoft Visual C++ 2010
  - Borland C++ Compiler 5.82
- An IDE
  - **Code Blocks<=Preferred**
  - Microsoft Visual C++ 2010 Express Studio
  - Turbo C++

- IDE= Integrated Development Environment
  - A set of programming tools for writing applications (source code editor, compiler, linker, debugger, etc.), all activated from a common user interface and menus.
- Open Code Blocks

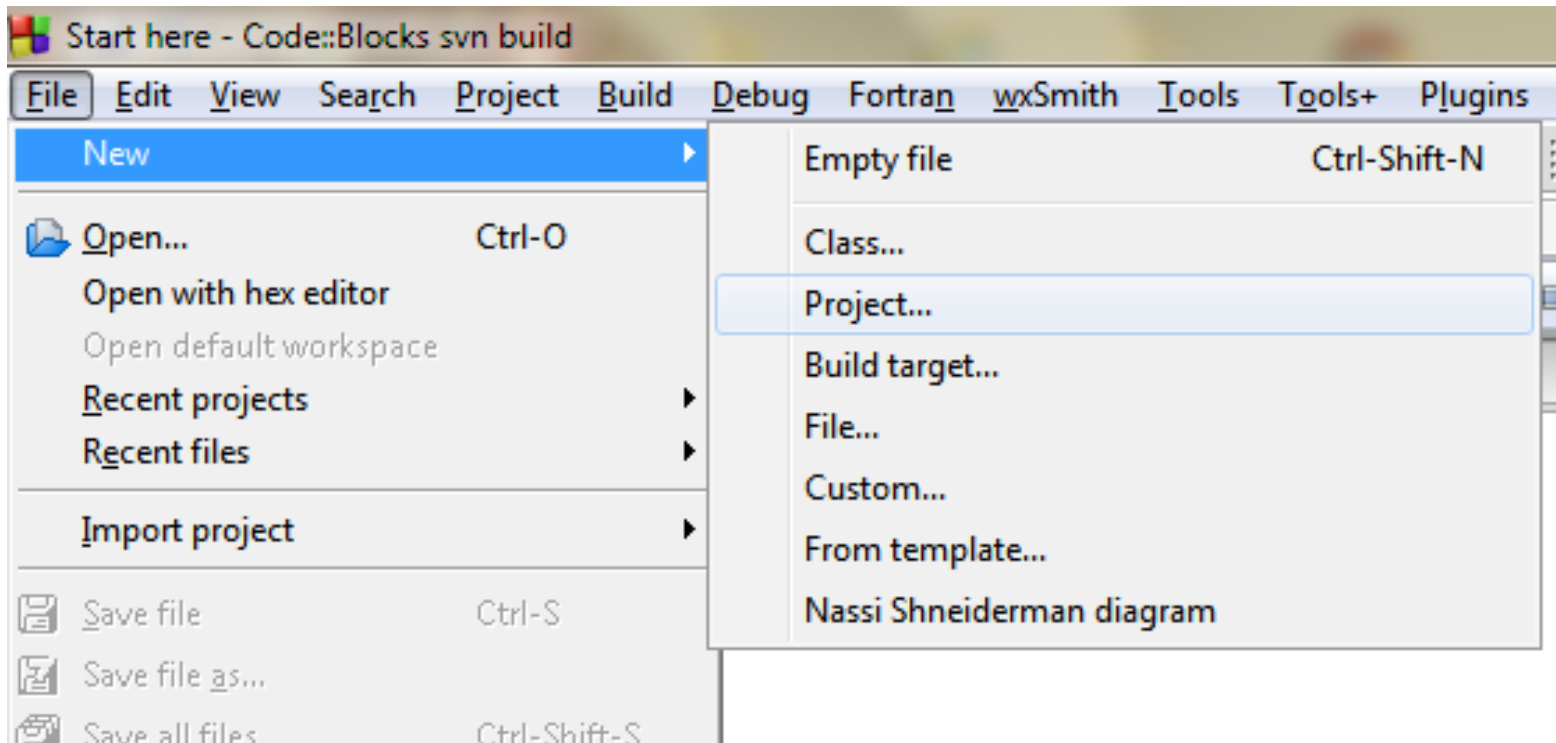


# The Environment

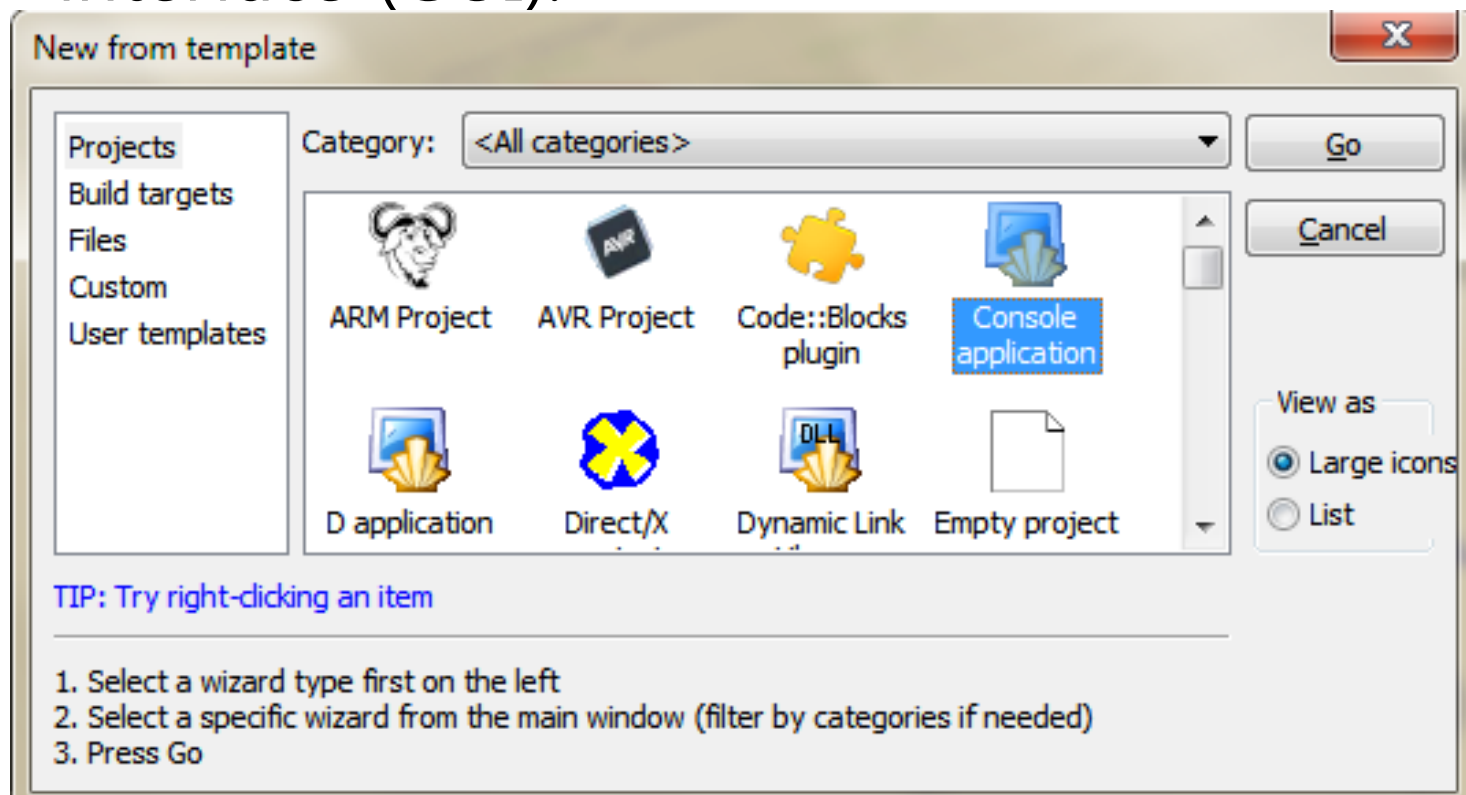


# Create a new Project

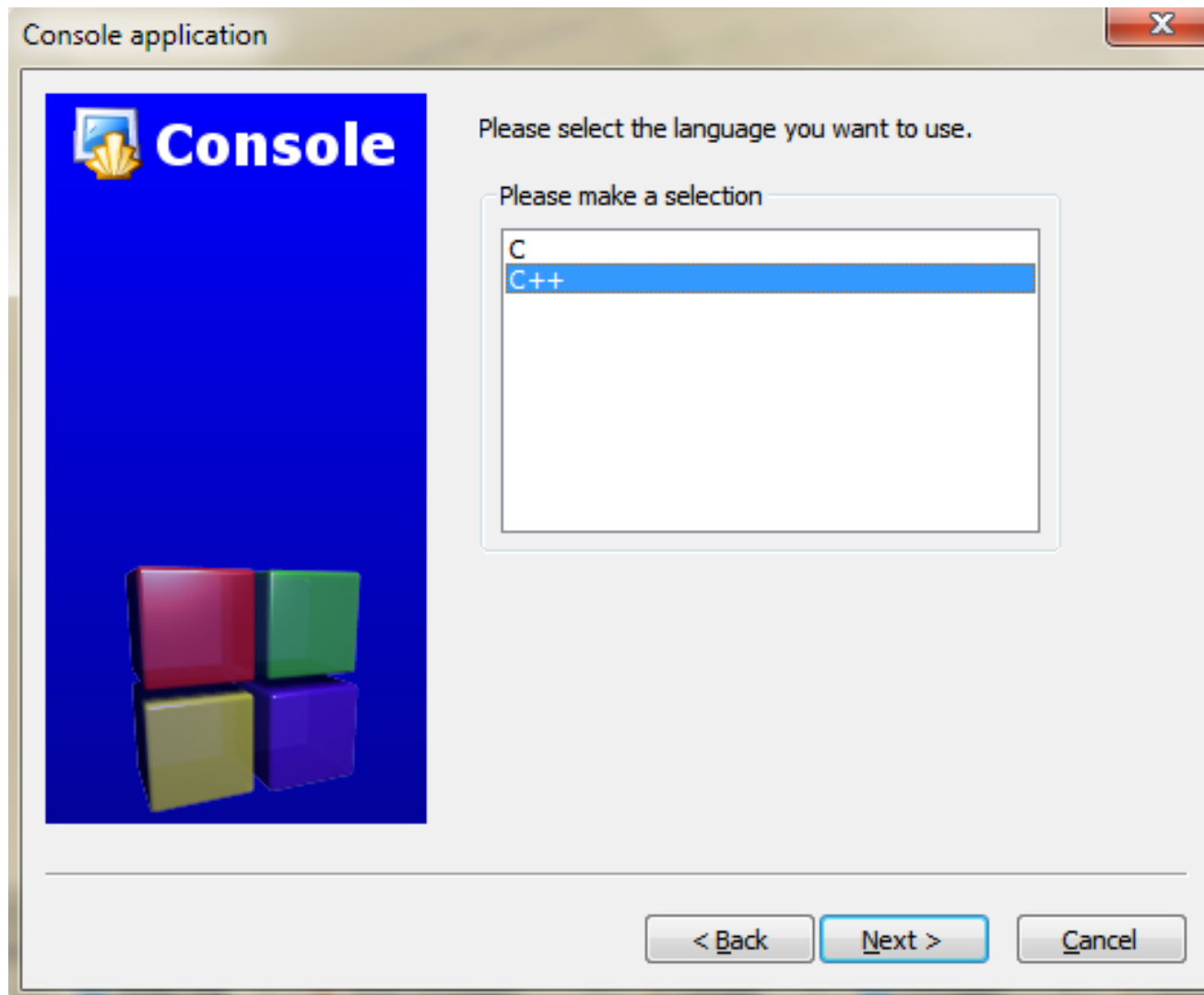
- An *application* is a runnable program. Most applications are built from multiple source files.
  - Code::Blocks keep track of the files and compiler settings need to build an application with a *project* file



- We use Console Applications
  - An application that uses the command line for input and output rather than a graphical interface (GUI).



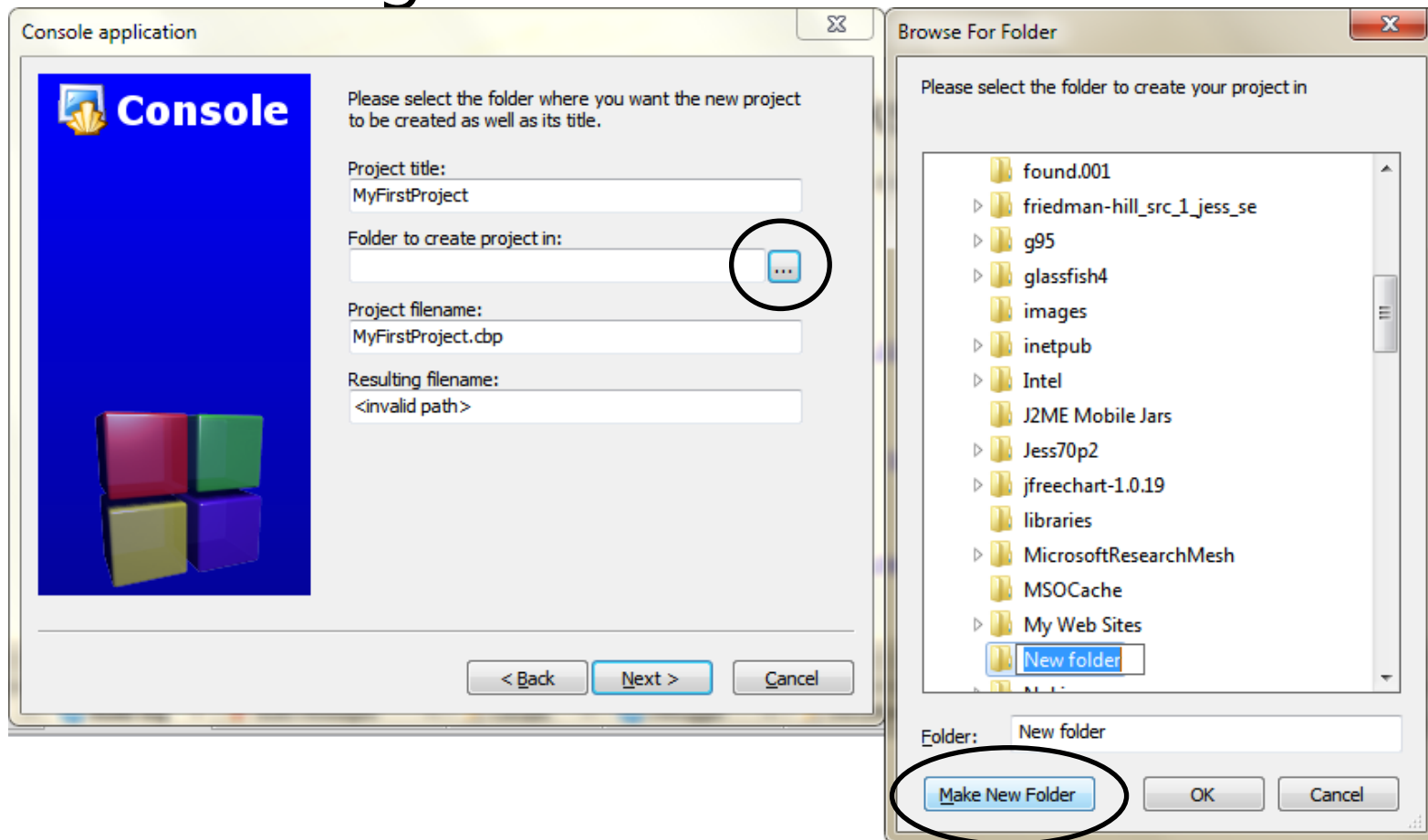
# The Console Application Wizard



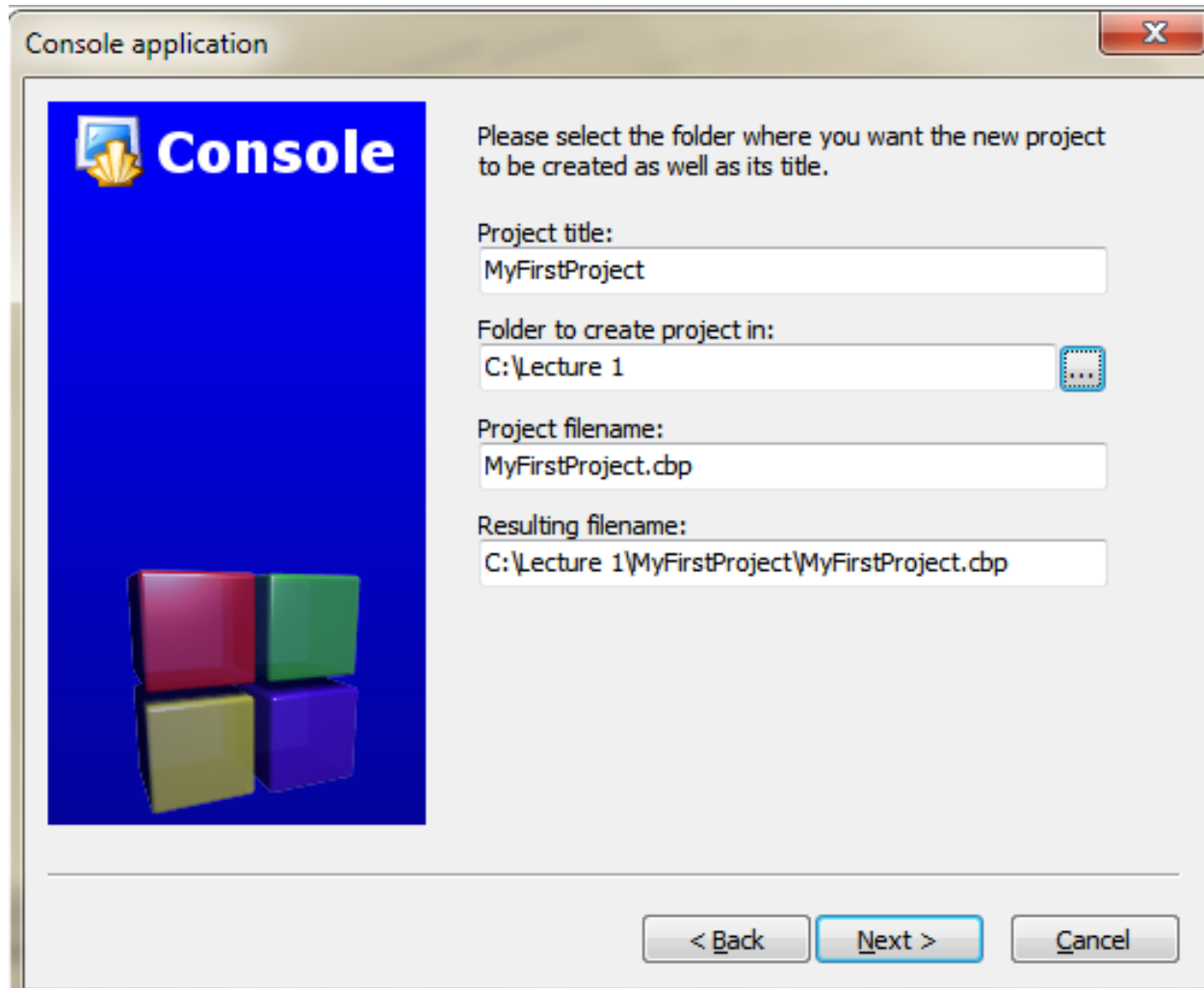


# Project Properties

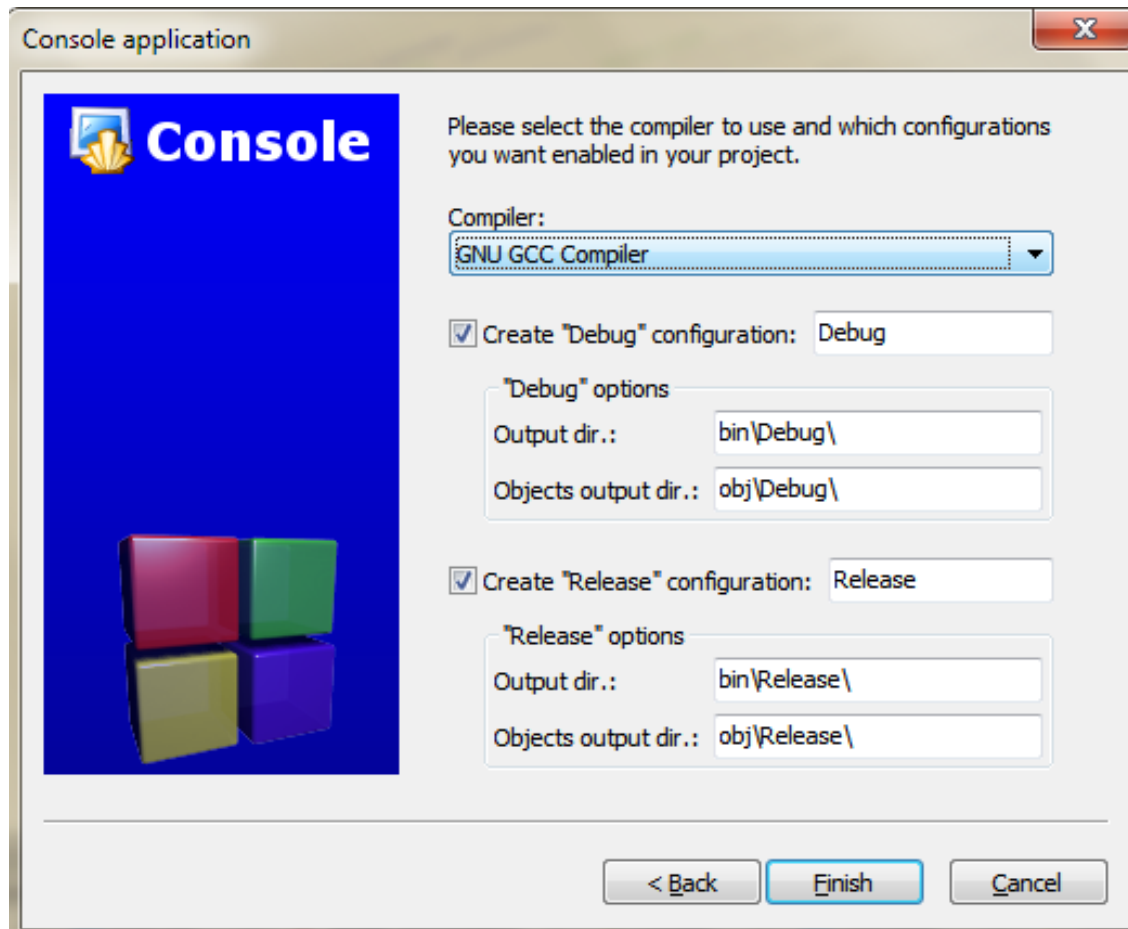
- Give descriptive project title
- Create a folder on c: drive(preferred) or use existing folder



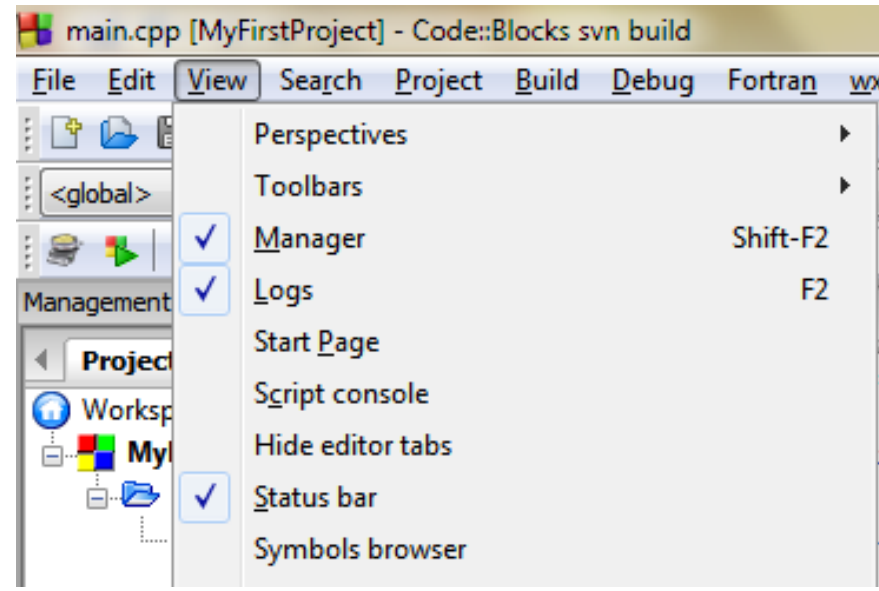
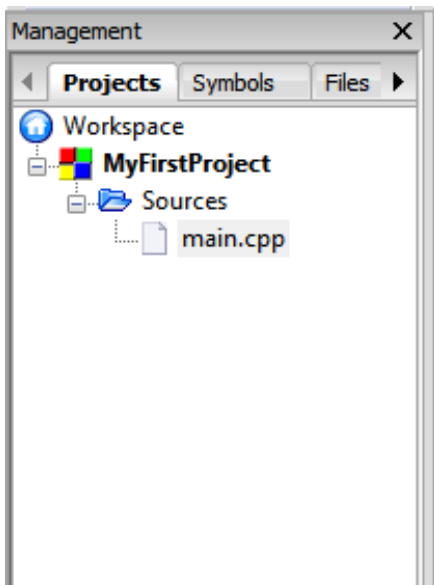
- Cbp=Code Blocks Project



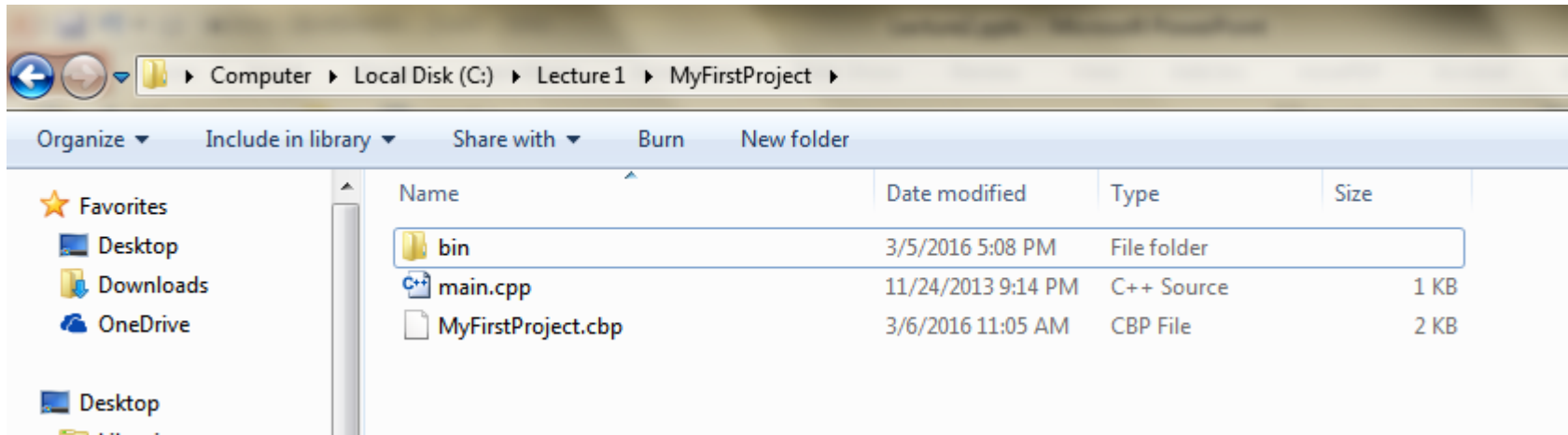
# Select Compiler



- Workspace collects projects together
  - where you can easily switch between projects
- The Projects are listed on the Manager
  - One sample c++ source file is inserted
- Can't see Manager?
  - make sure it is checked on the view menu

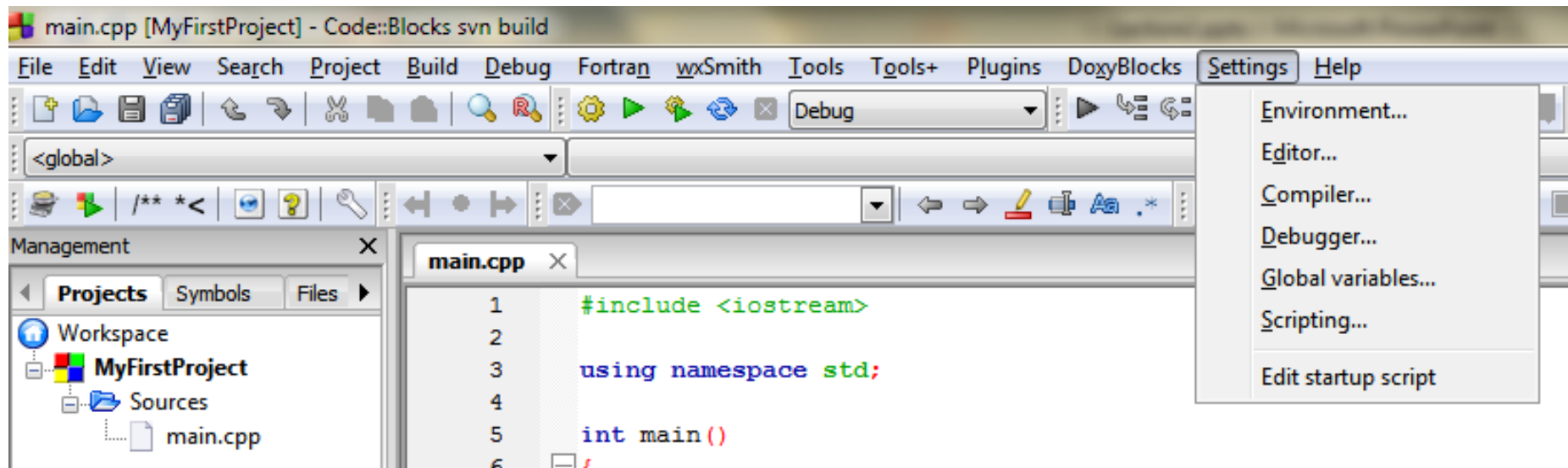


- View your Project in Windows Explorer

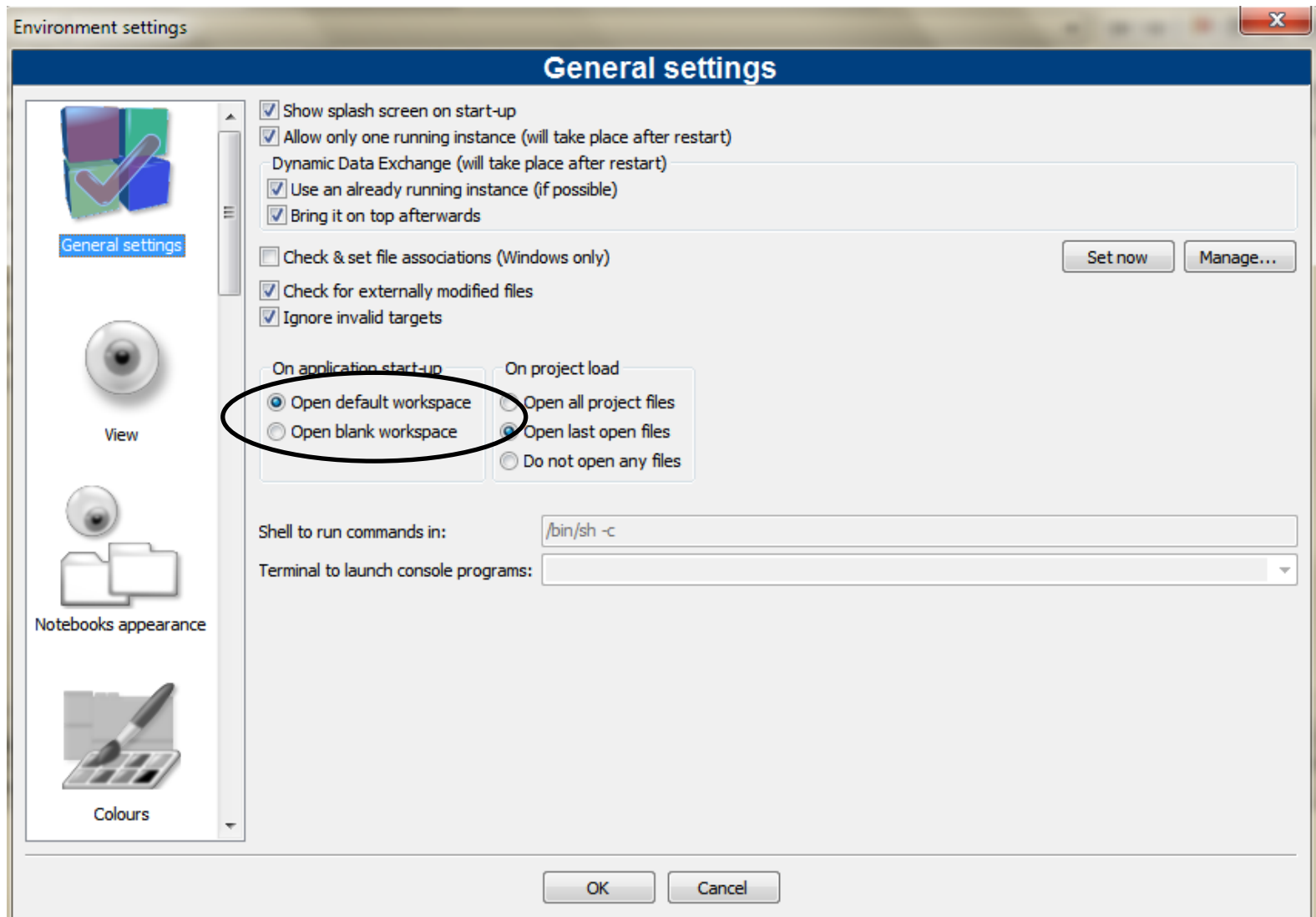


# Settings

- We can change settings as desired

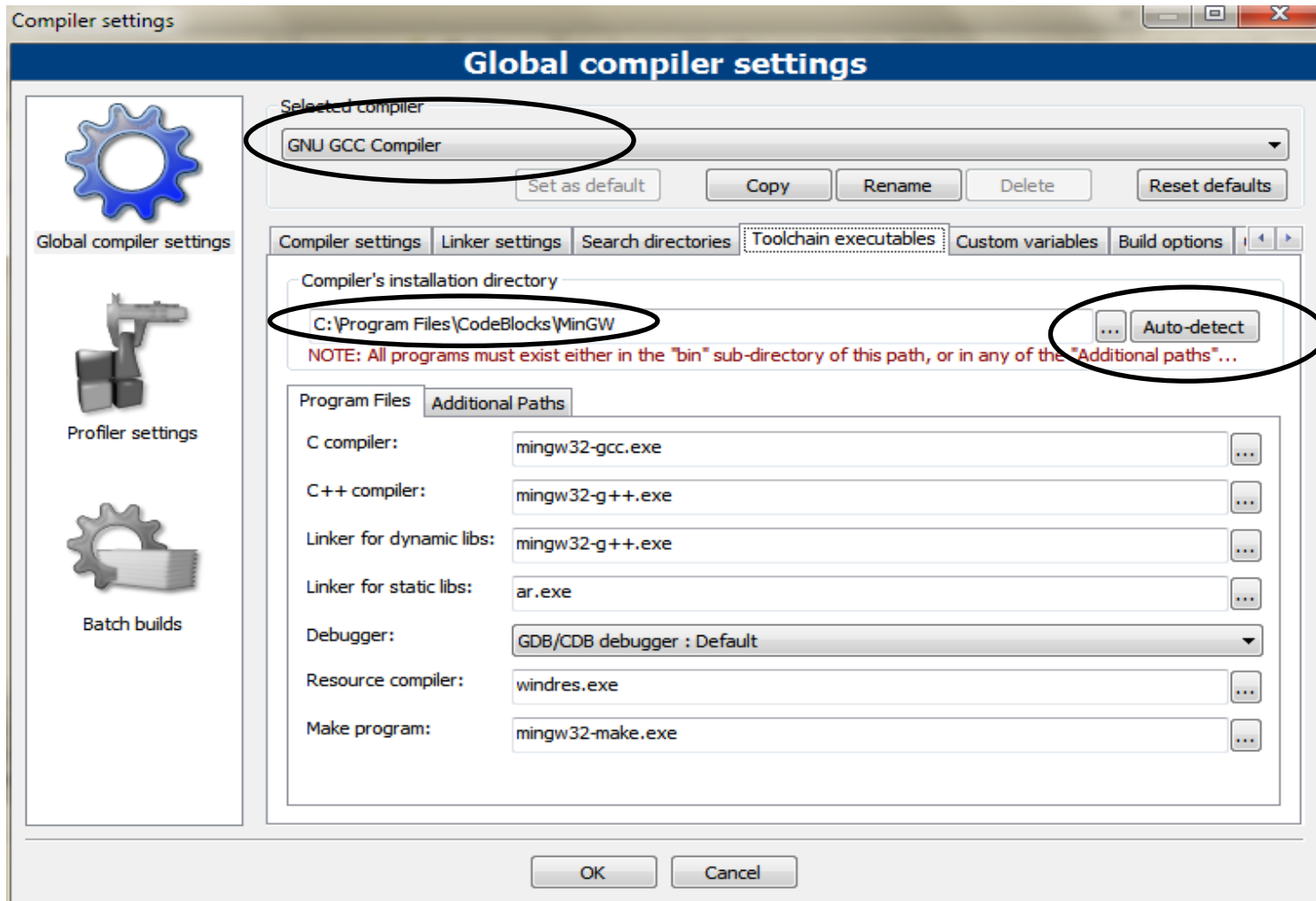


# Environment Setting



# Compiler Settings

- Don't change this if you didn't install additional compilers separately





# Our First C++ Program

---

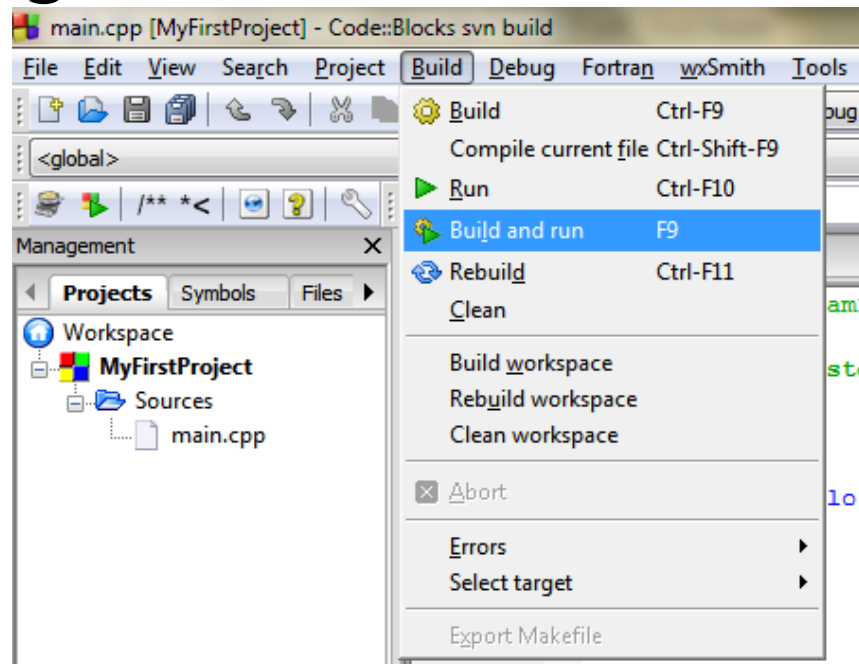
```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Hello world!" << endl;
8      return 0;
9  }
10
```

- A namespace is a collection of name definitions.
  - One name, such as a function name, can be given different definitions in two namespaces.
  - A program can then use one of these namespaces in one place and the other in another location.

- All the standard libraries we will be using place their definitions in the **std(standard) namespace**.
  - It is almost impossible to write a C++ program without using at least one of these libraries.
- To use any of these definitions in your program, you must insert the following using directive:  
**using namespace std;**

- For example, the library for console I/O is **`iostream`**. So, most of our demonstration programs will begin  
**`#include <iostream>`**

- **Build:** The source program will be compiled and linker links the object files created by the compiler into an executable
- **Run:** The program will be in memory



# Console Screen

- We can change the color and Font of the console
  - Right click on the title bar
  - Choose properties

