C++

Prepared by: Mohamed Ayman

facebook.com/sw.eng.MohamedAyman

sw.eng.MohamedAyman@gmail.com

wuzzuf.net/me/engMohamedAyman

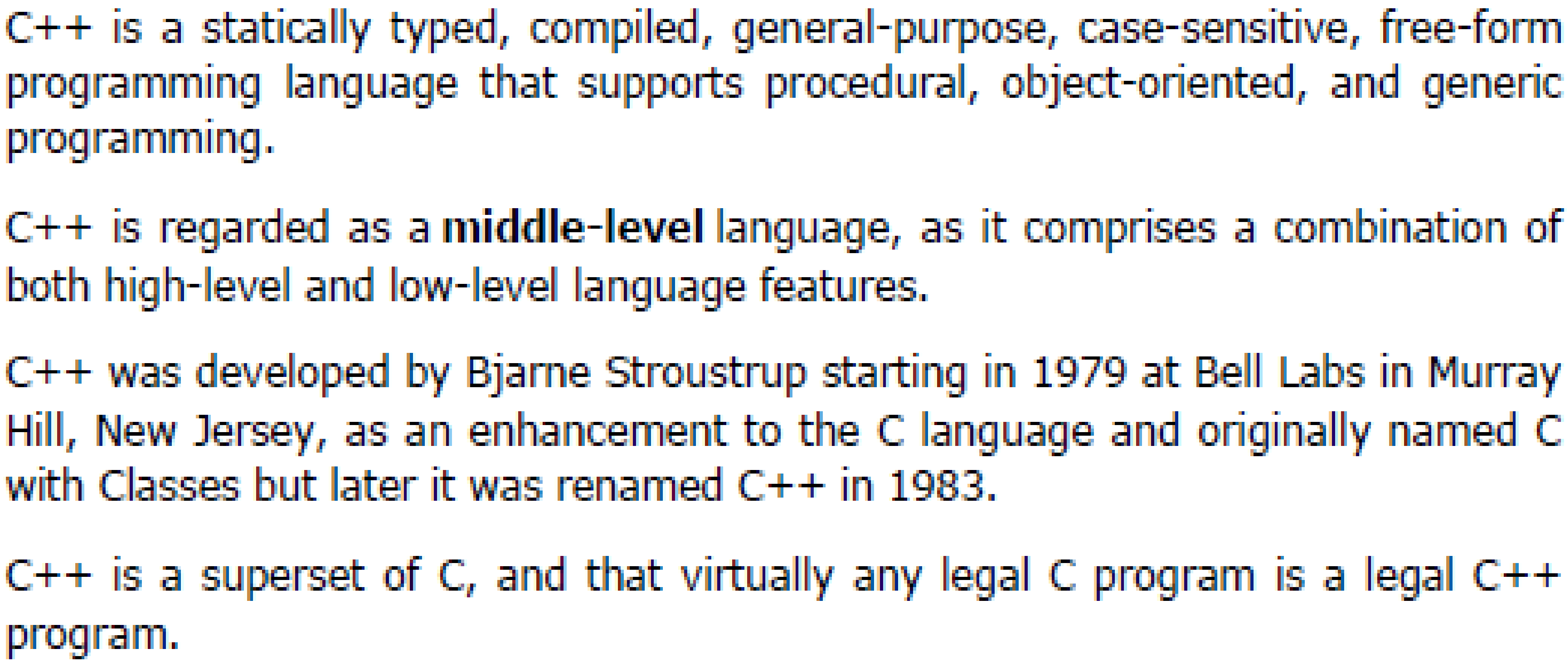
codeforces.com/profile/Mohamed_Ayman

Overview

Outline

1) History of C++
2) C++ Features
3) Interpreter vs. Compiler
4) C++ Compiler (Download, Installation)
5) First C++ Program
6) C++ Identifiers
7) Reserved Words
8) Comments in C++

History of C++



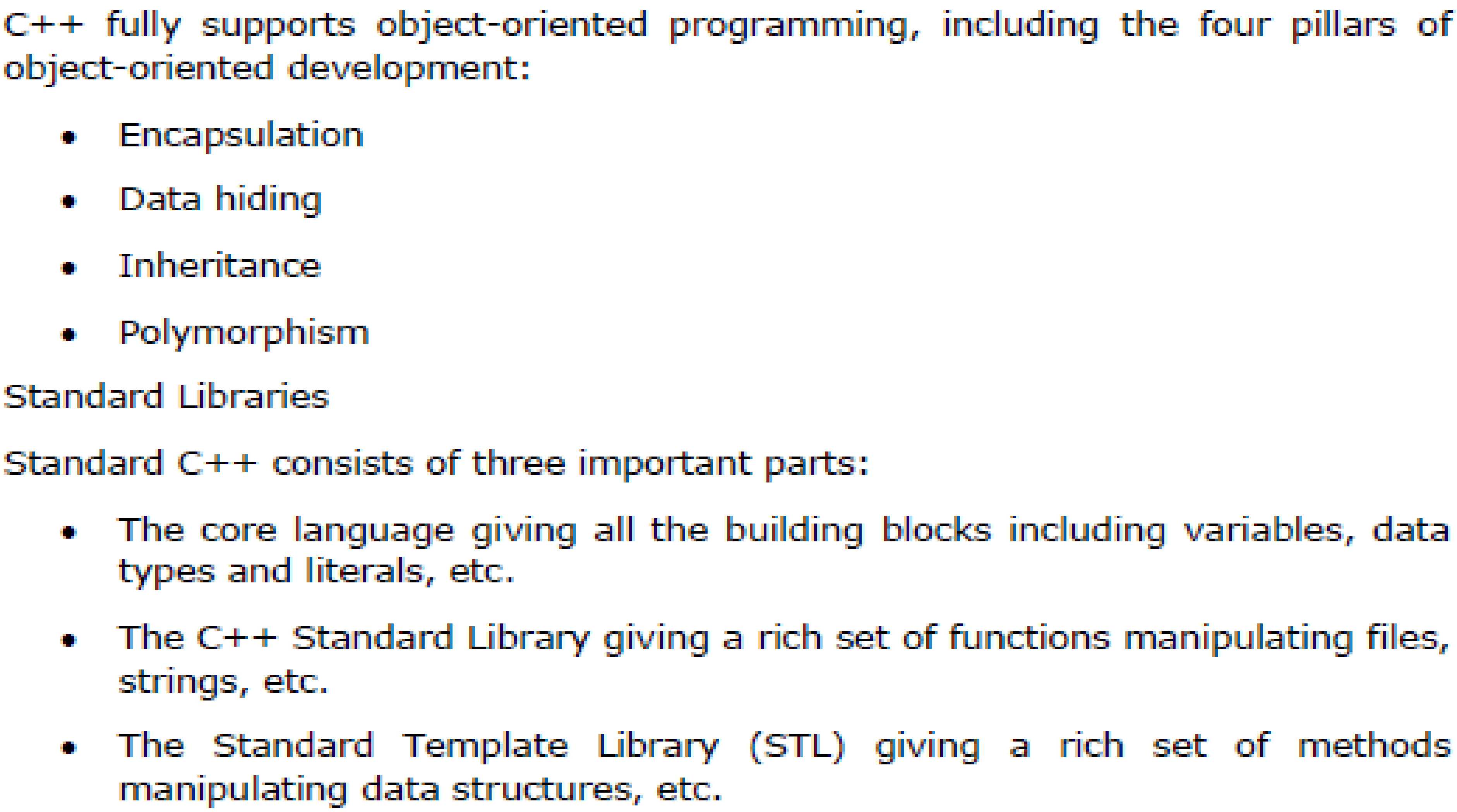
C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.

C++ is regarded as a **middle-level** language, as it comprises a combination of both high-level and low-level language features.

C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.

C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

C++ Features



C++ fully supports object-oriented programming, including the four pillars of object-oriented development:

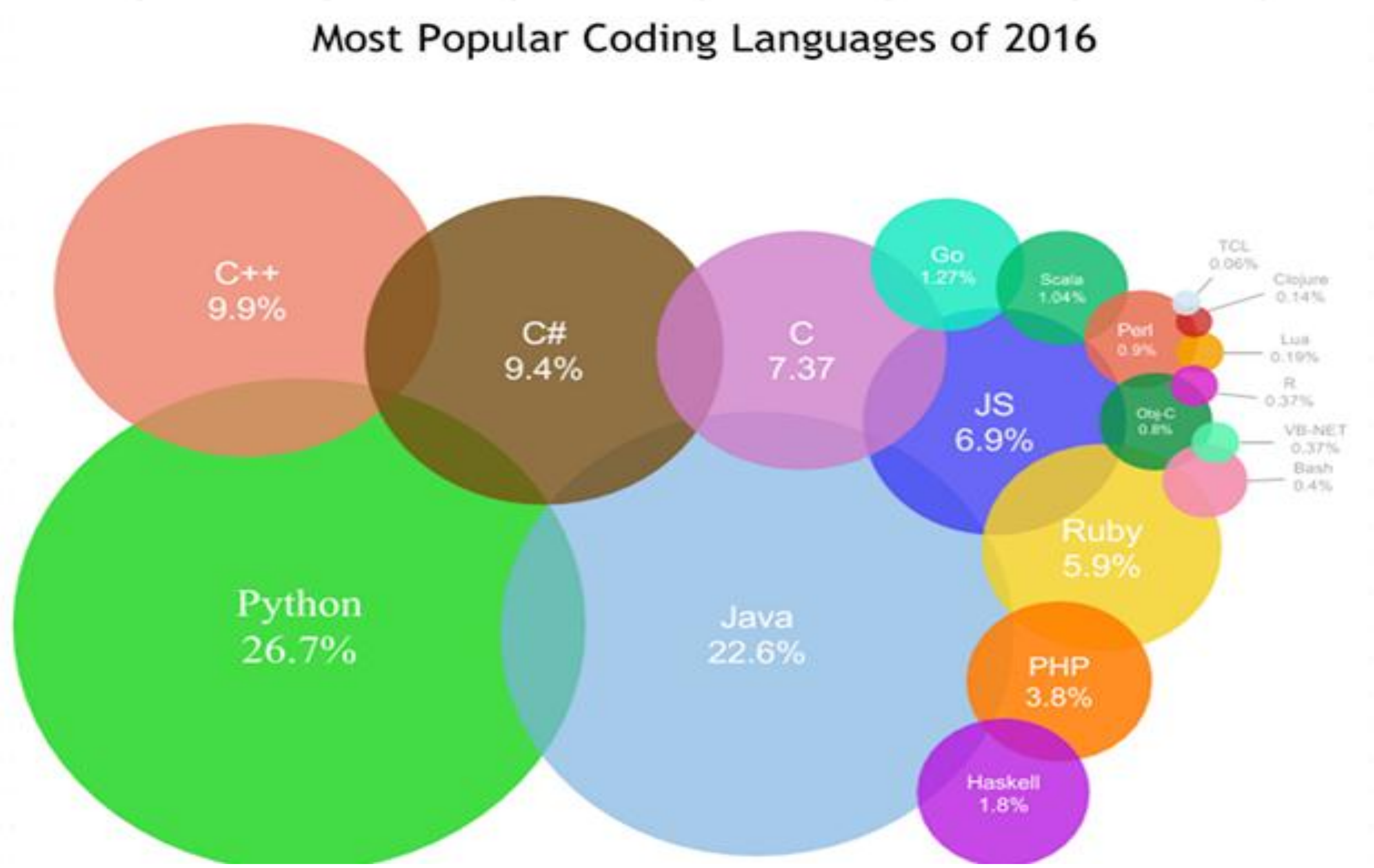
- Encapsulation
- Data hiding
- Inheritance
- Polymorphism

Standard Libraries

Standard C++ consists of three important parts:

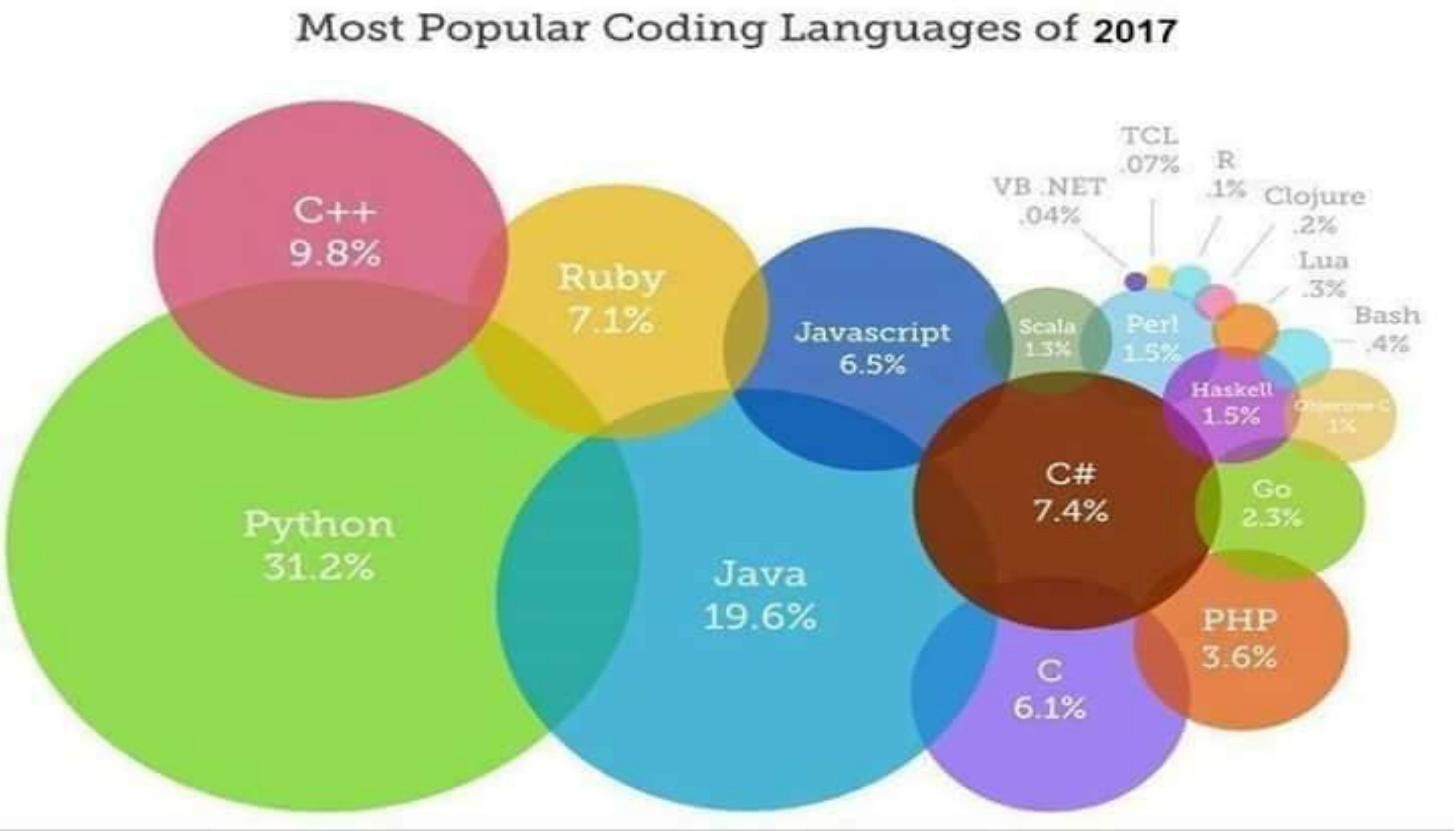
- The core language giving all the building blocks including variables, data types and literals, etc.
- The C++ Standard Library giving a rich set of functions manipulating files, strings, etc.
- The Standard Template Library (STL) giving a rich set of methods manipulating data structures, etc.

C++ in 2016



Most Popular Coding Languages of 2016

C++ in 2017



Most Popular Coding Languages of 2017

Programming languages

- Low-level languages:

  machine oriented and require extensive knowledge of computer hardware and its configuration

- Types:
  - machine language
  - assembly language
- High-level Language:

  programming language that uses English and mathematical symbols in its instructions

- Types:
  - compiled
  - interpreted

Types of Low-level languages

- machine language:

  language that is directly understood by the computer,

  and it does not need to be translated

- assembly language:

  set of symbols and letters

Types of High-level languages

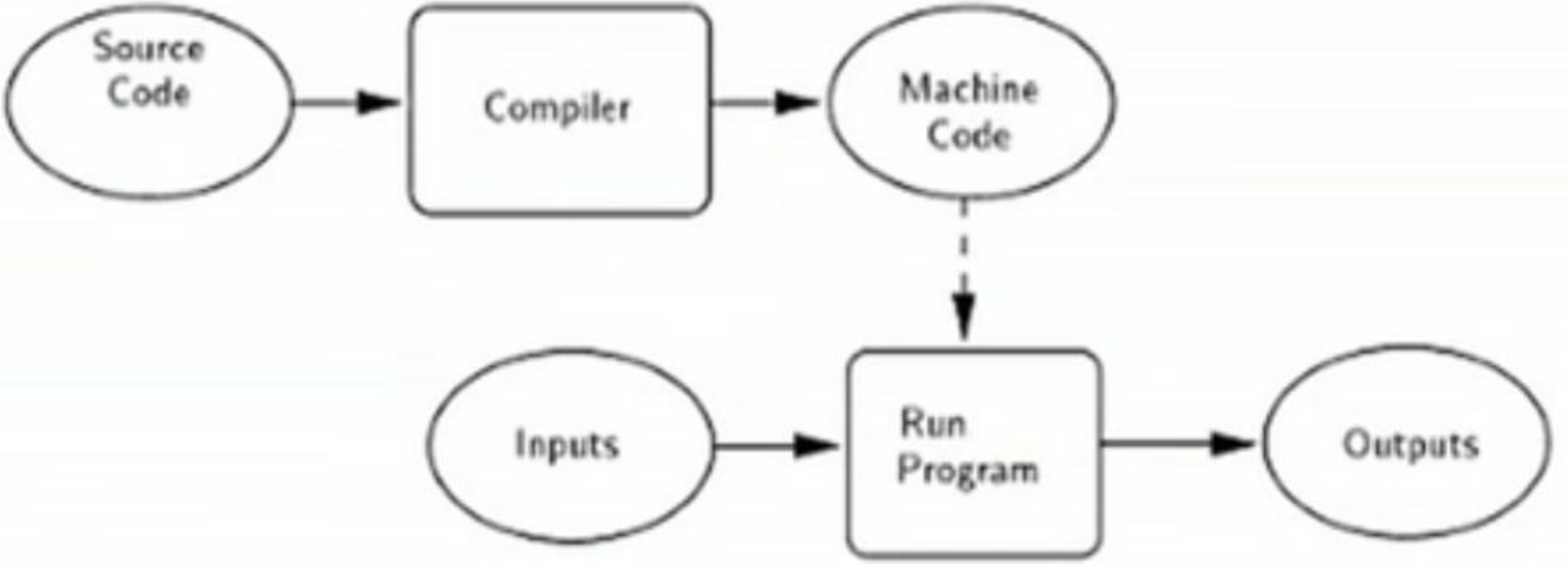
- Compiled:

computer program that translates a program written in a high-level language to the machine language of a computer

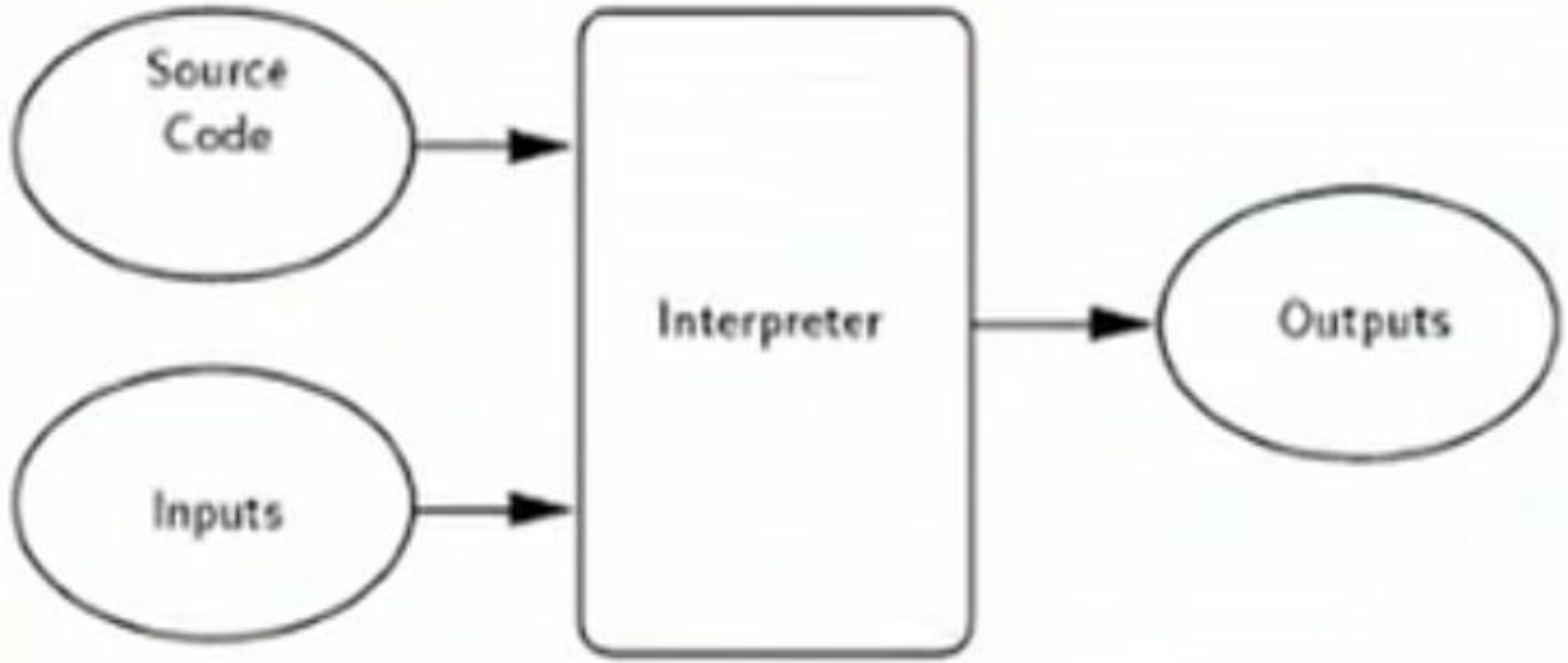
- Interpreted:

computer program that simulates a computer that understands a high-level language

Interpreter vs. Compiler



Interpreter vs. Compiler



C++ Online Compiler

- Tutorials point online compiler

tutorialspoint.com/compile_cpp_online.php

- repl.it online compiler

repl.it/languages/cpp11

- ideone online compiler

ideone.com

C++ Code Blocks Compiler

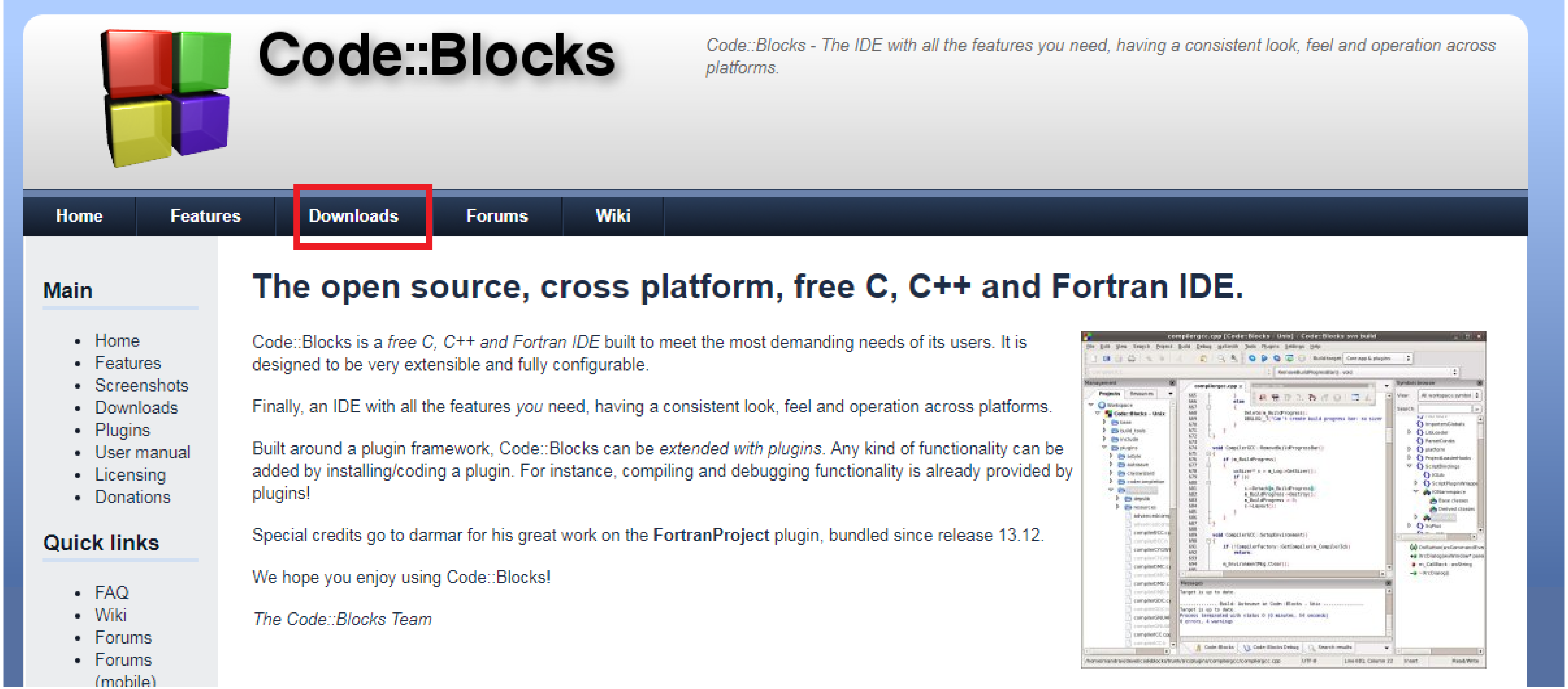
- Code Blocks website

http://www.codeblocks.org/

- Installing Code Blocks Video

https://www.youtube.com/watch?v=aS5_jrlbKmA

C++ Compiler Code Blocks Download



Code::Blocks - The IDE with all the features you need, having a consistent look, feel and operation across platforms.

Main

- Home
- Features
- Screenshots
- Downloads
- Plugins
- User manual
- Licensing
- Donations

Quick links

- FAQ
- Wiki
- Forums
- Forums (mobile)

The open source, cross platform, free C, C++ and Fortran IDE.

Code::Blocks is a *free C, C++ and Fortran IDE* built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Finally, an IDE with all the features *you* need, having a consistent look, feel and operation across platforms.

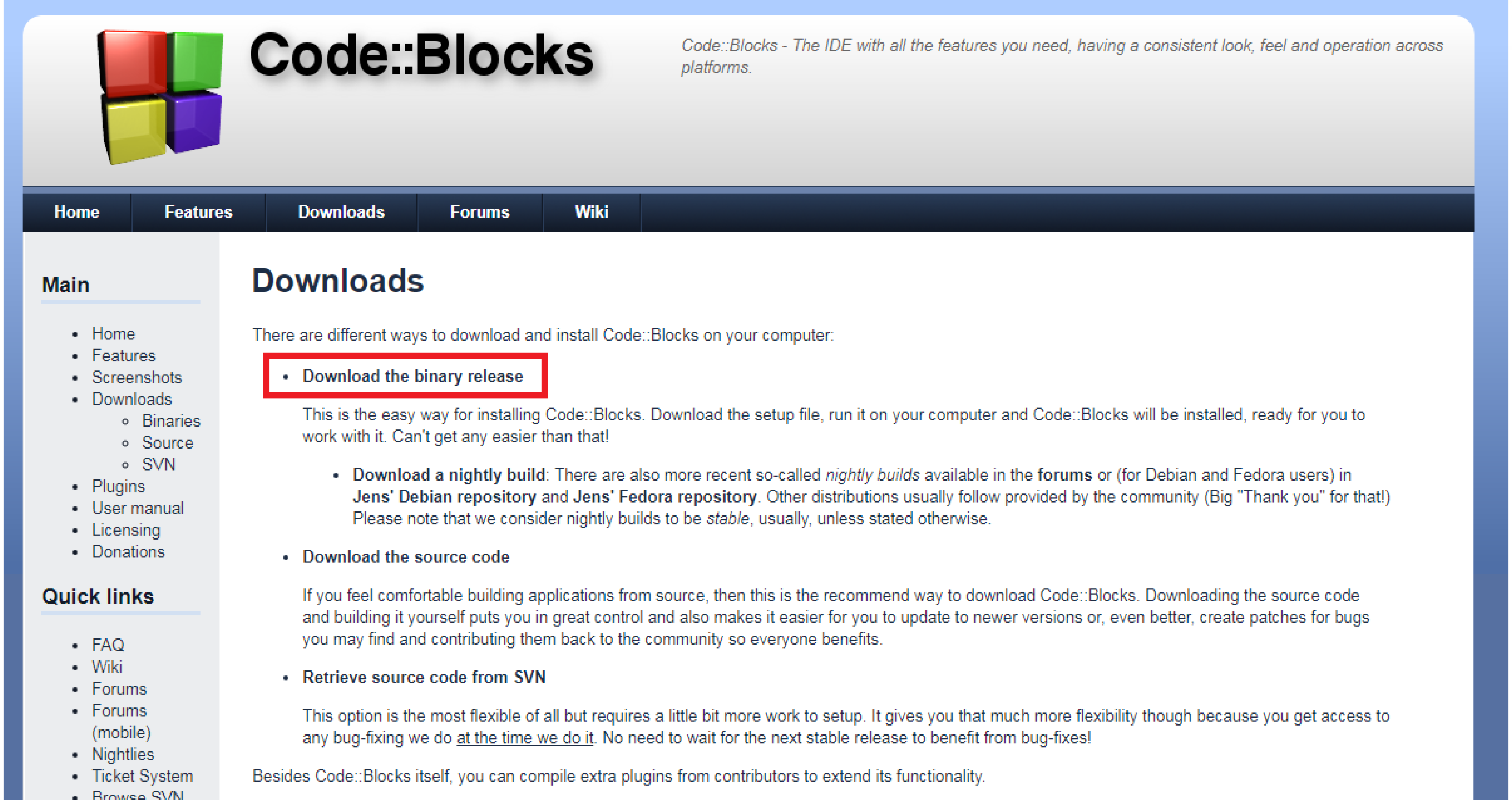
Built around a plugin framework, Code::Blocks can be *extended with plugins*. Any kind of functionality can be added by installing/coding a plugin. For instance, compiling and debugging functionality is already provided by plugins!

Special credits go to darmar for his great work on the **FortranProject** plugin, bundled since release 13.12.

We hope you enjoy using Code::Blocks!

The Code::Blocks Team

C++ Compiler Code Blocks Download



Code::Blocks - The IDE with all the features you need, having a consistent look, feel and operation across platforms.

Main

- Home
- Features
- Screenshots
- Downloads
  - Binaries
  - Source
  - SVN
- Plugins
- User manual
- Licensing
- Donations

Quick links

- FAQ
- Wiki
- Forums
- Forums (mobile)
- Nightlies
- Ticket System
- Browse SVN

Downloads

There are different ways to download and install Code::Blocks on your computer:

- **Download the binary release**

  This is the easy way for installing Code::Blocks. Download the setup file, run it on your computer and Code::Blocks will be installed, ready for you to work with it. Can't get any easier than that!

  - **Download a nightly build**: There are also more recent so-called *nightly builds* available in the **forums** or (for Debian and Fedora users) in **Jens' Debian repository** and **Jens' Fedora repository**. Other distributions usually follow provided by the community (Big "Thank you" for that!) Please note that we consider nightly builds to be *stable*, usually, unless stated otherwise.

- **Download the source code**

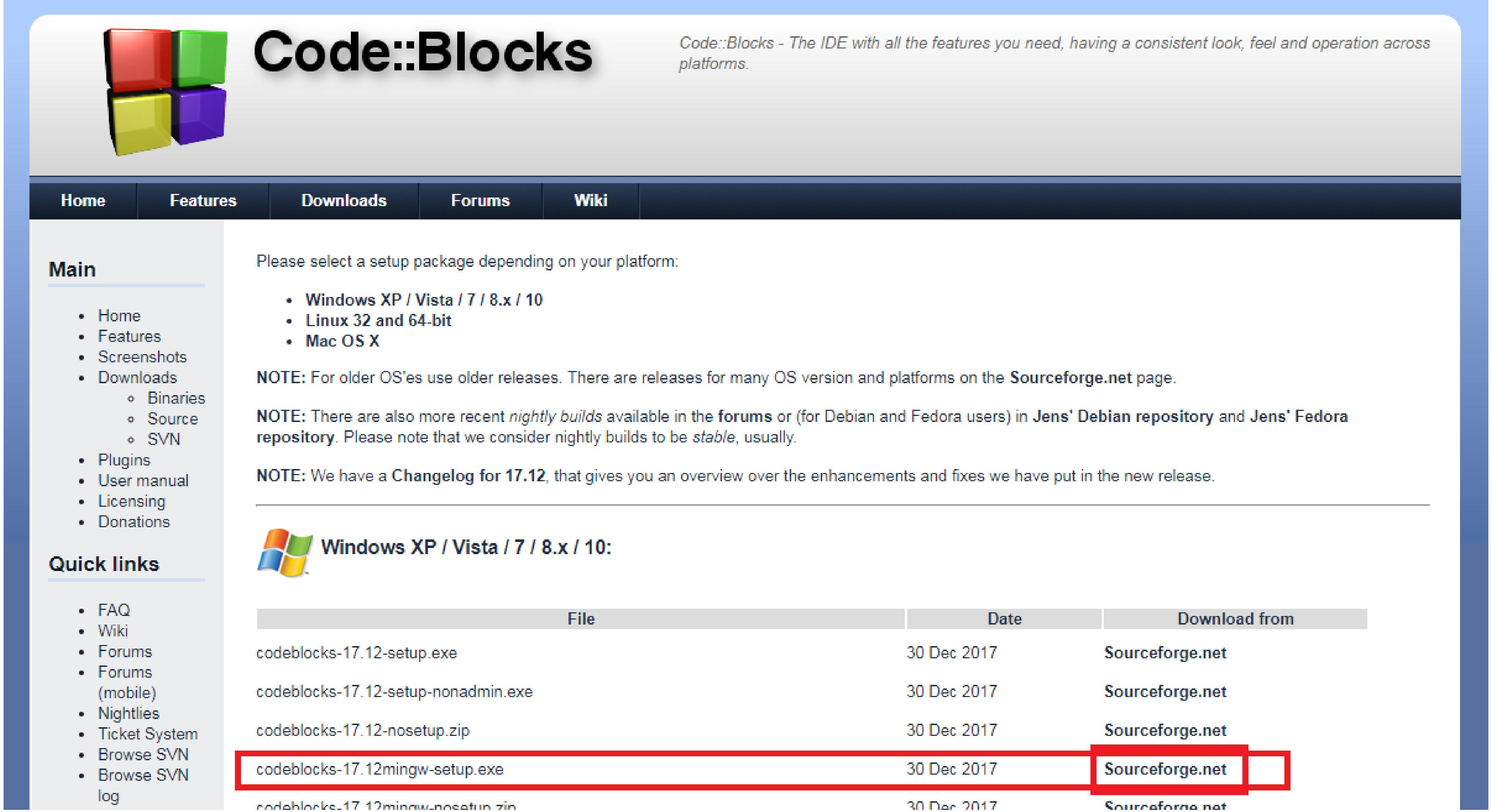
  If you feel comfortable building applications from source, then this is the recommend way to download Code::Blocks. Downloading the source code and building it yourself puts you in great control and also makes it easier for you to update to newer versions or, even better, create patches for bugs you may find and contributing them back to the community so everyone benefits.

- **Retrieve source code from SVN**

  This option is the most flexible of all but requires a little bit more work to setup. It gives you that much more flexibility though because you get access to any bug-fixing we do at the time we do it. No need to wait for the next stable release to benefit from bug-fixes!

Besides Code::Blocks itself, you can compile extra plugins from contributors to extend its functionality.

C++ Compiler Code Blocks Download



Please select a setup package depending on your platform:

- **Windows XP / Vista / 7 / 8.x / 10**
- **Linux 32 and 64-bit**
- **Mac OS X**

NOTE: For older OS'es use older releases. There are releases for many OS version and platforms on the **Sourceforge.net** page.

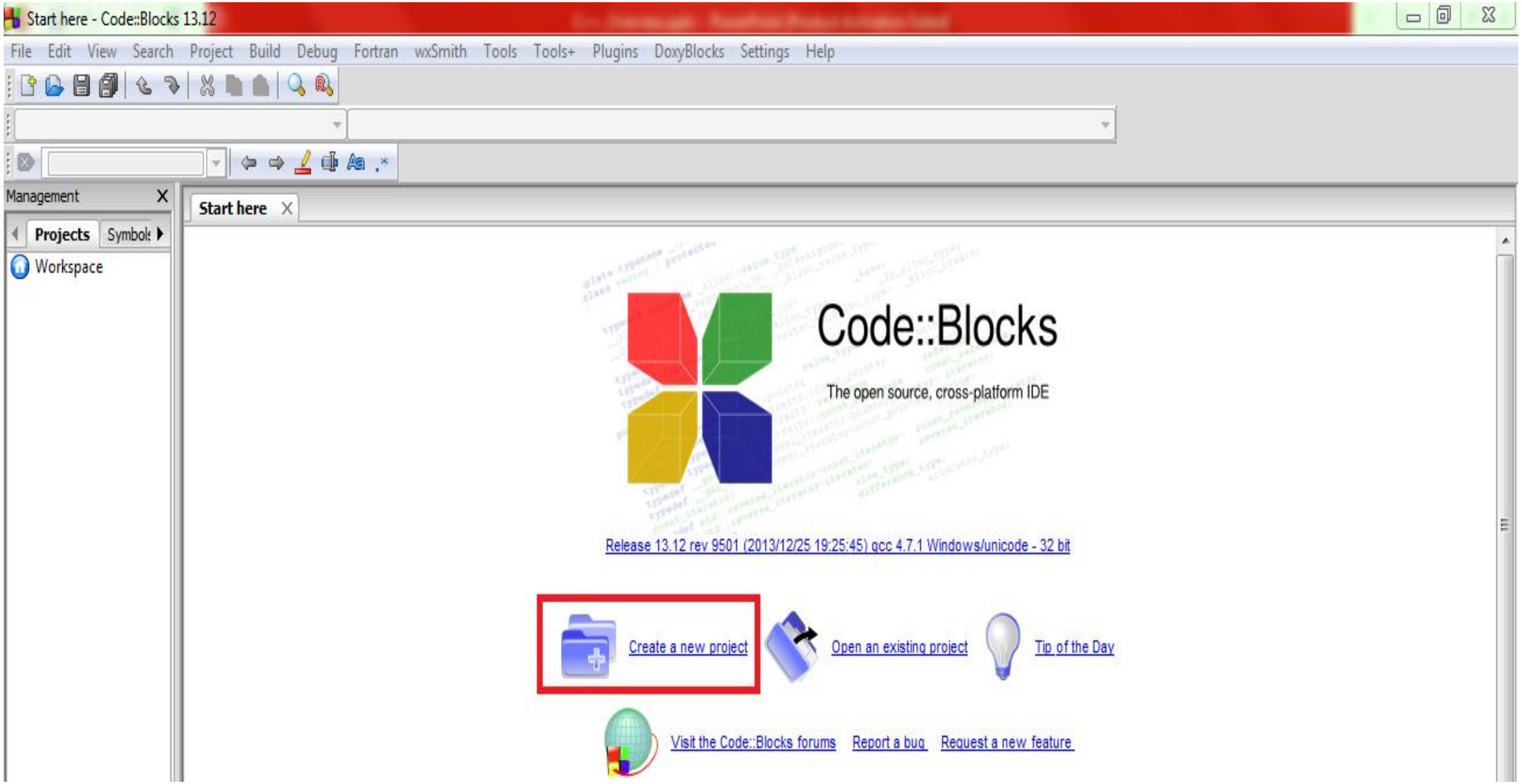
NOTE: There are also more recent *nightly builds* available in the **forums** or (for Debian and Fedora users) in **Jens' Debian repository** and **Jens' Fedora repository**. Please note that we consider nightly builds to be *stable*, usually.

NOTE: We have a **Changelog for 17.12**, that gives you an overview over the enhancements and fixes we have put in the new release.

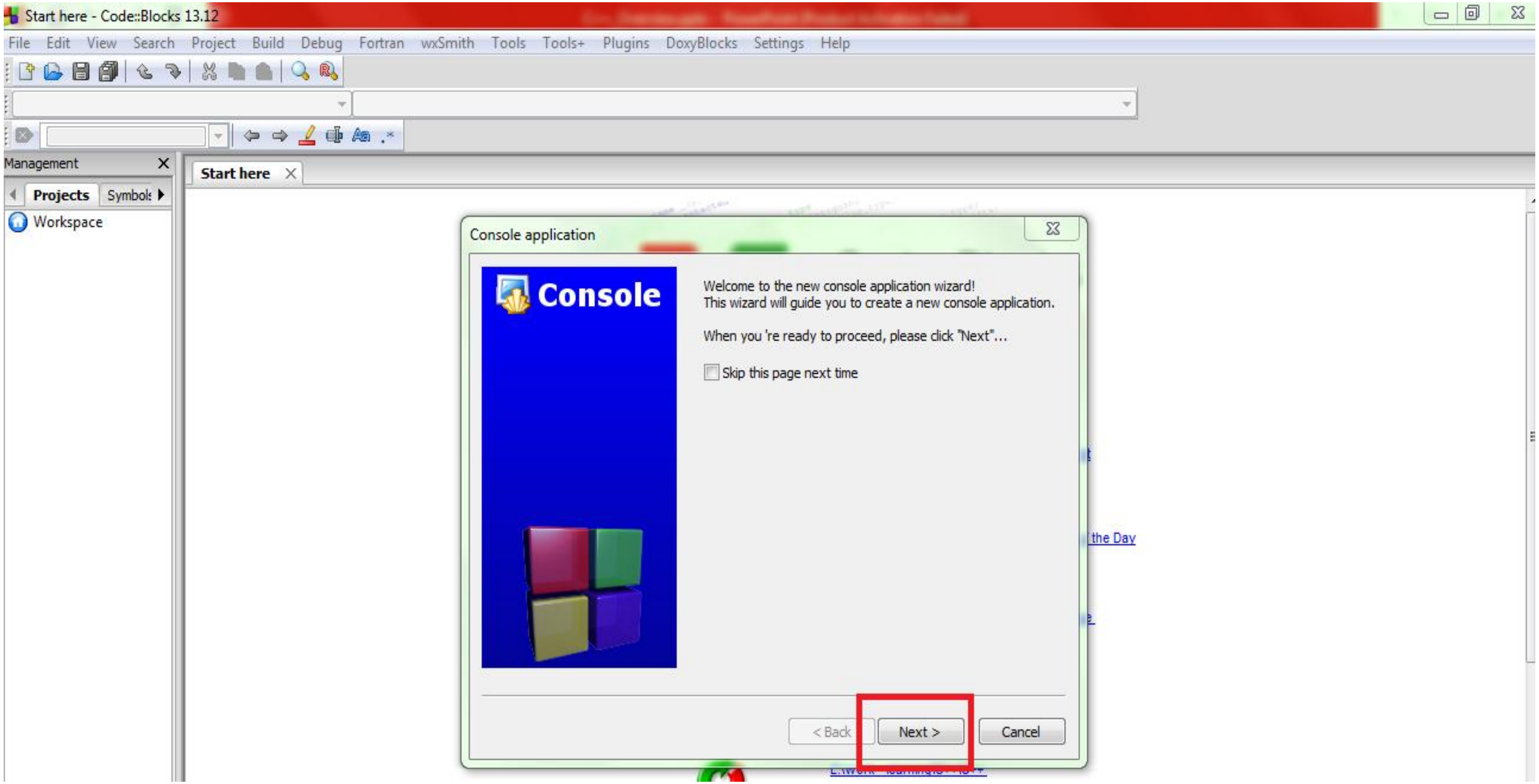
Windows XP / Vista / 7 / 8.x / 10:

File	Date	Download from
codeblocks-17.12-setup.exe	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12-setup-nonadmin.exe	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12-nosetup.zip	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12mingw-setup.exe	30 Dec 2017	**Sourceforge.net**
codeblocks-17.12mingw-nosetup.zip	30 Dec 2017	**Sourceforge.net**

C++ Compiler Code Blocks Running


Start here - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols
Workspace
Start here
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

C++ Compiler Code Blocks Running



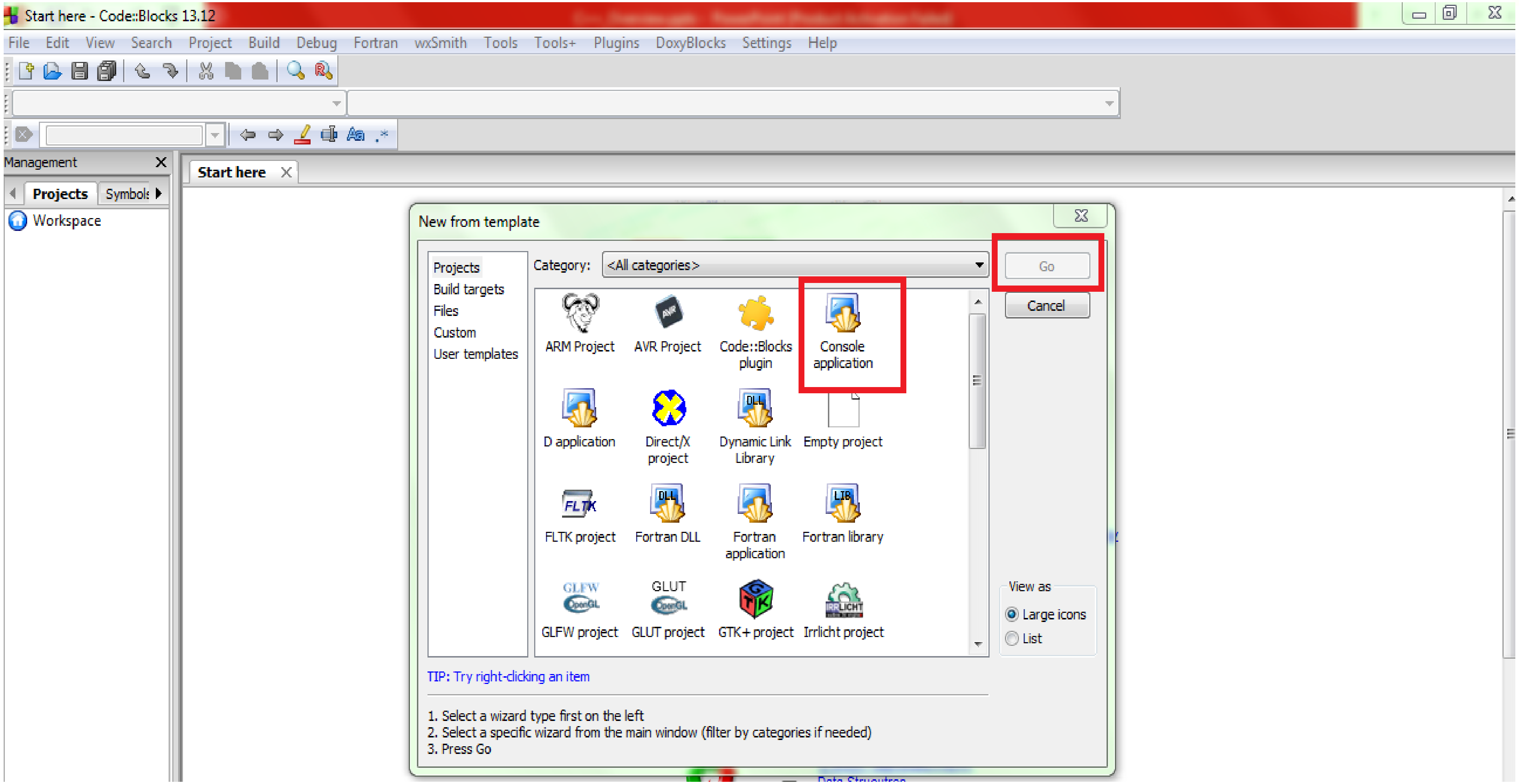
Welcome to the new console application wizard!
This wizard will guide you to create a new console application.

When you 're ready to proceed, please click "Next"...

□ Skip this page next time

Start here - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects
Symbols
Workspace
Start here
Console application
Console
< Back
Next >
Cancel
the Day

C++ Compiler Code Blocks Running



Start here - Code::Blocks 13.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects
Symbols
Workspace
Start here

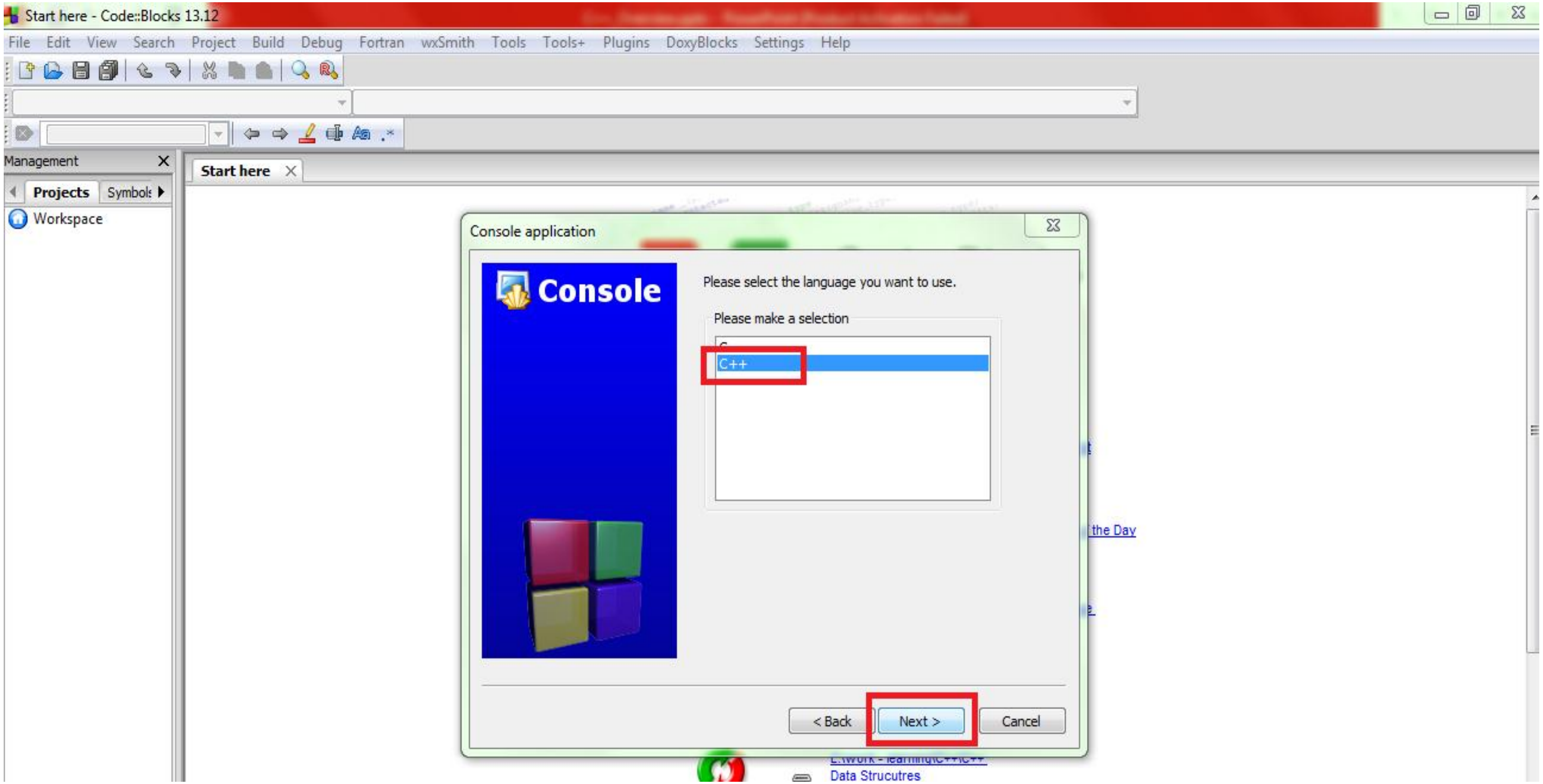
New from template

Category: <All categories>

TIP: Try right-clicking an item

1. Select a wizard type first on the left
2. Select a specific wizard from the main window (filter by categories if needed)
3. Press Go

C++ Compiler Code Blocks Running



Start here - Code::Blocks 13.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects
Symbols
Workspace
Start here
Console application
Console

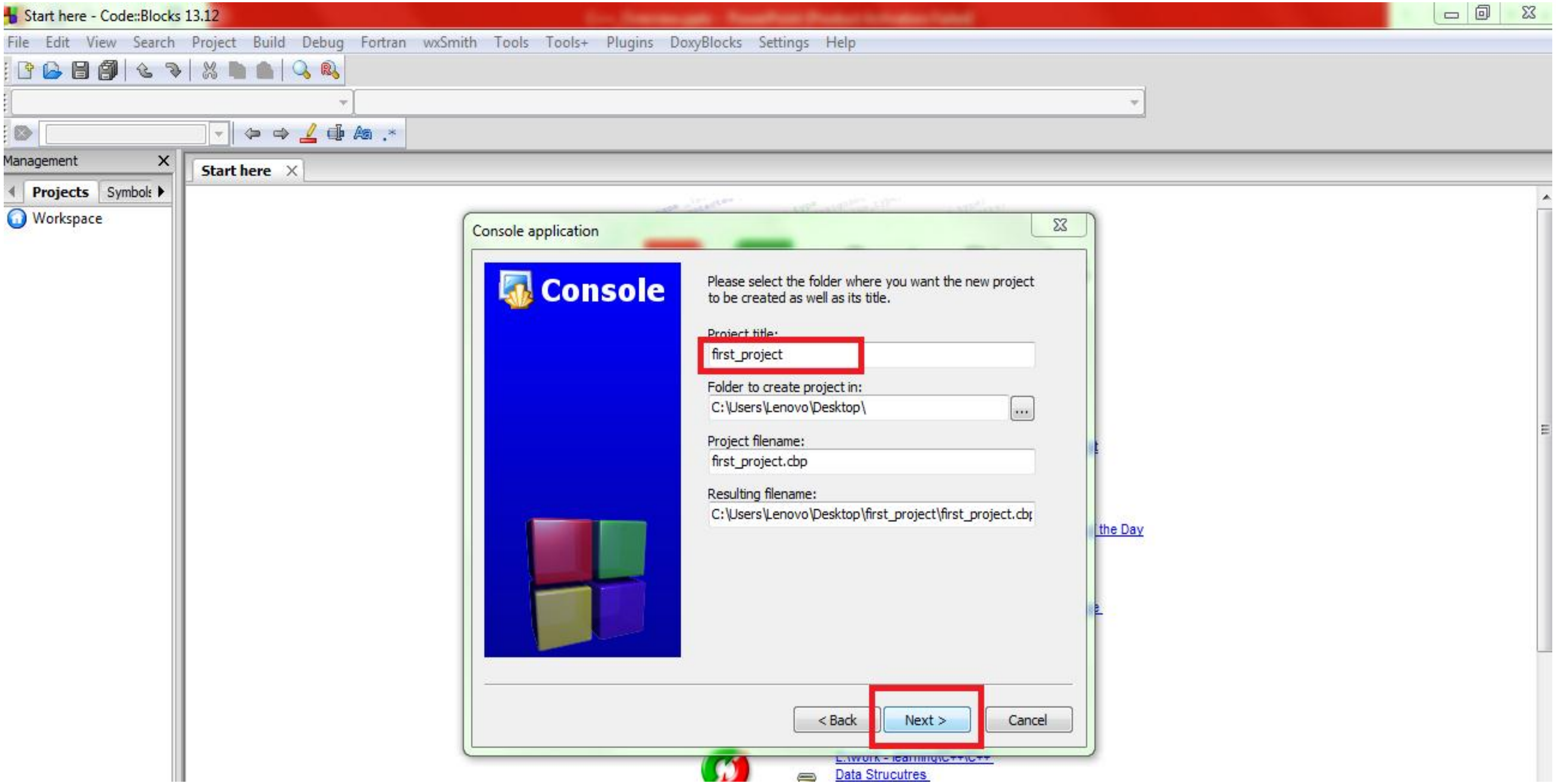
Please select the language you want to use.

Please make a selection

C

C++

C++ Compiler Code Blocks Running


Start here - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects
Symbols
Workspace
Start here
Console
the Day
Data Strucutres
< Back
Next >
Cancel

Console application

Please select the folder where you want the new project to be created as well as its title.

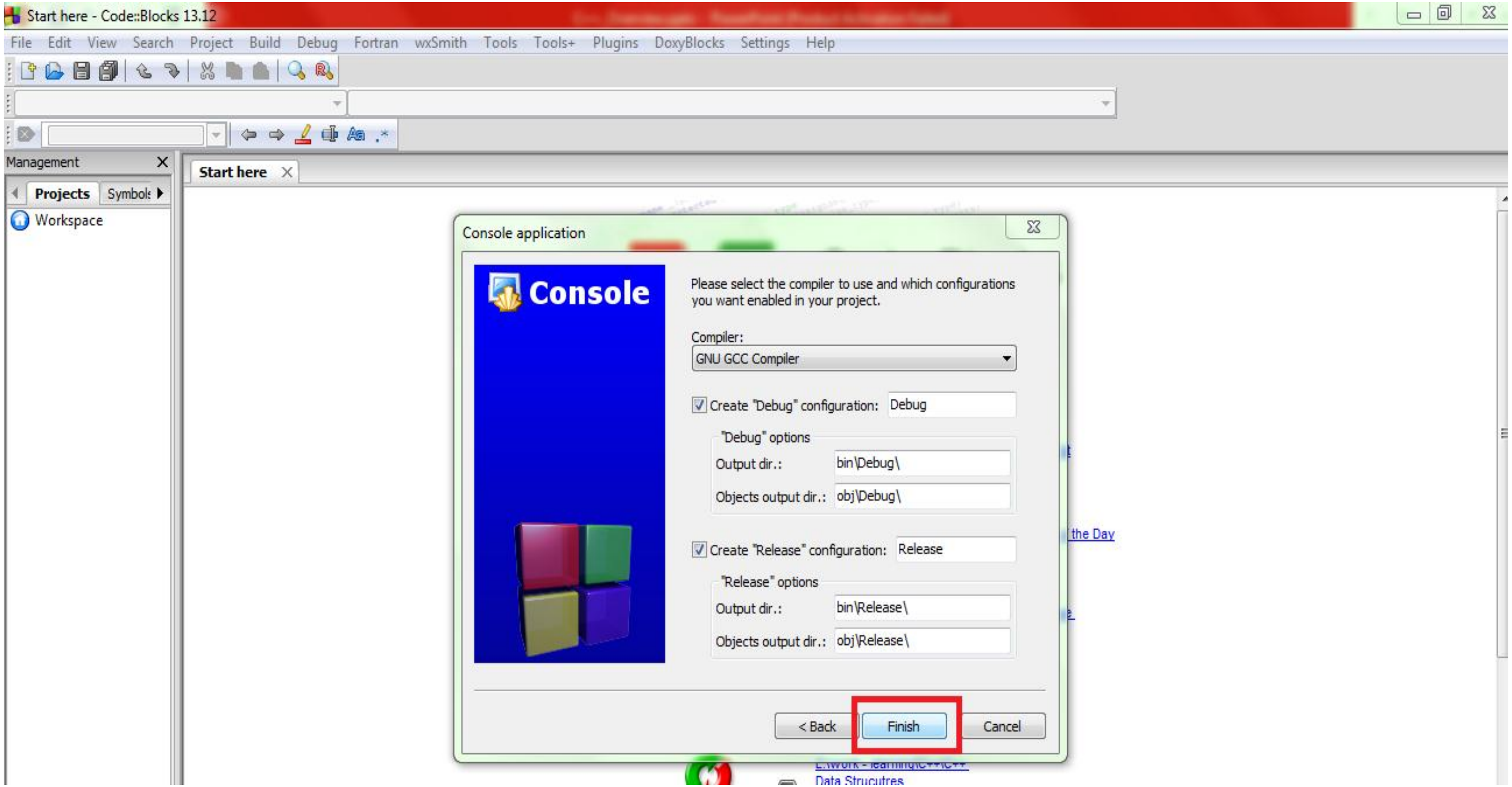
Project title: first_project

Folder to create project in: C:\Users\Lenovo\Desktop\

Project filename: first_project.cbp

Resulting filename: C:\Users\Lenovo\Desktop\first_project\first_project.cbp

C++ Compiler Code Blocks Running



Start here - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects
Symbols
Workspace
Start here
Console
the Day
Data Strucutres

Console application

Please select the compiler to use and which configurations you want enabled in your project.

Compiler: GNU GCC Compiler

☑ Create "Debug" configuration: Debug

"Debug" options

Output dir.: bin\Debug\

Objects output dir.: obj\Debug\

☑ Create "Release" configuration: Release

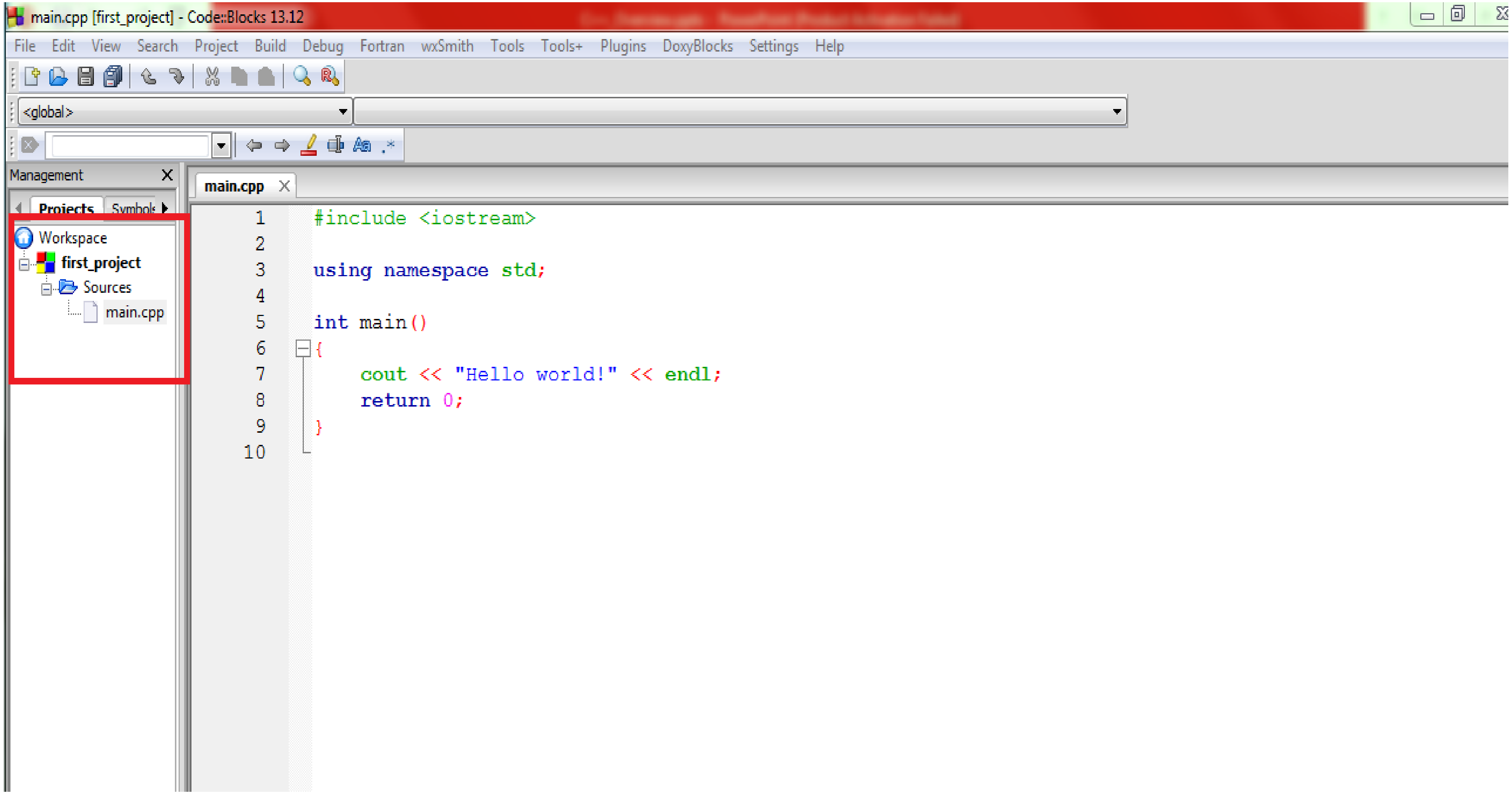
"Release" options

Output dir.: bin\Release\

Objects output dir.: obj\Release\

< Back | Finish | Cancel

C++ Compiler Code Blocks Running

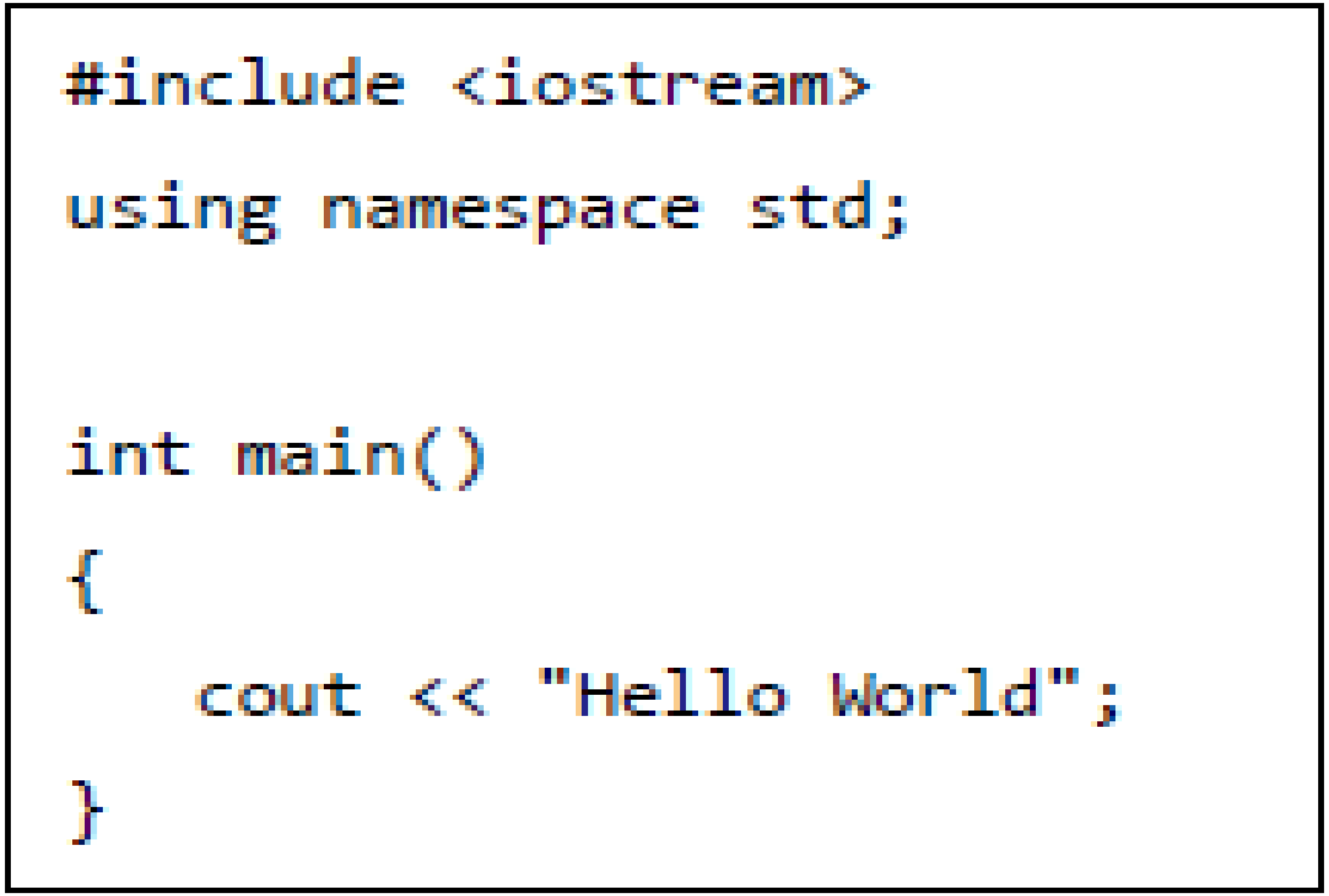

main.cpp [first_project] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Projects Symbols
Workspace
first_project
Sources
main.cpp
main.cpp

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

First C++ Program



```
#include <iostream>
using namespace std;


int main()
{
    cout << "Hello World";
}
```

First C++ Program

- Let us look at the various parts of the above program:

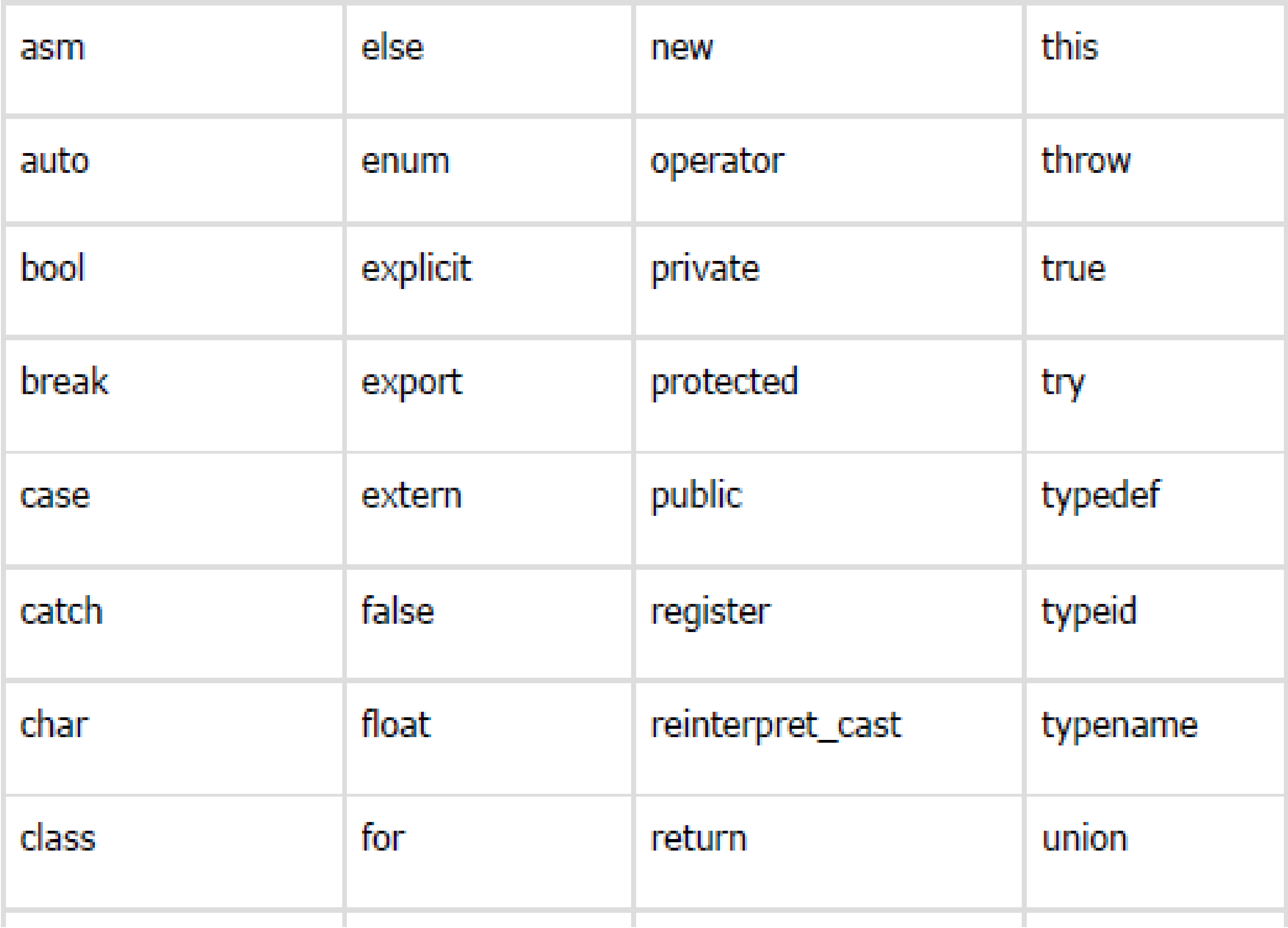
1. The C++ language defines several headers, which contain information that is either necessary or useful to your program. For this program, the header <iostream> is needed.
2. The line using namespace std; tells the compiler to use the std namespace. Namespaces are a relatively recent addition to C++.
3. The line int main() is the main function where program execution begins.
4. The next line cout << “Hello World”; causes the message “Hello World” to be displayed on the screen.
5. In C++, the semicolon is a statement terminator. That is, each individual statement must be ended with a semicolon. It indicates the end of one logical entity.

C++ Identifiers

- A C++ identifier is a name used to identify a variable, function, class, module or other objects.
- An identifier starts with a letter [A to Z] or [a to z] or an underscore (_) followed by zero or more letters, underscore and digits (0 to 9).
- C++ does not allow punctuation characters such as @, $, and % within identifiers.
- C++ is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in C++.

Reserved Words

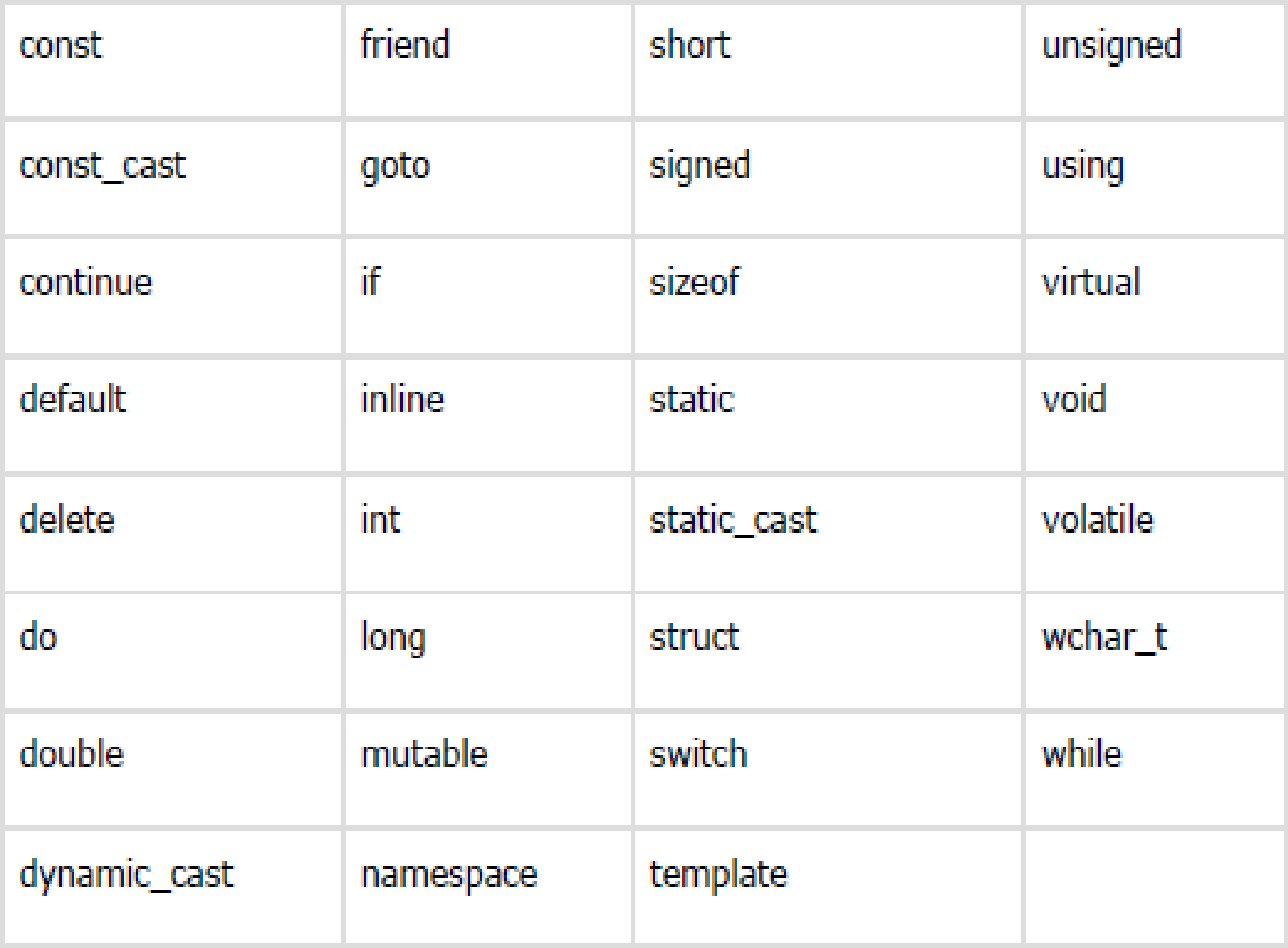
- The following list shows the C++ keywords. These are reserved words and you cannot use them as constants or variables or any other identifier name.



asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union

Reserved Words

- The following list shows the C++ keywords. These are reserved words and you cannot use them as constants or variables or any other identifier name.



const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

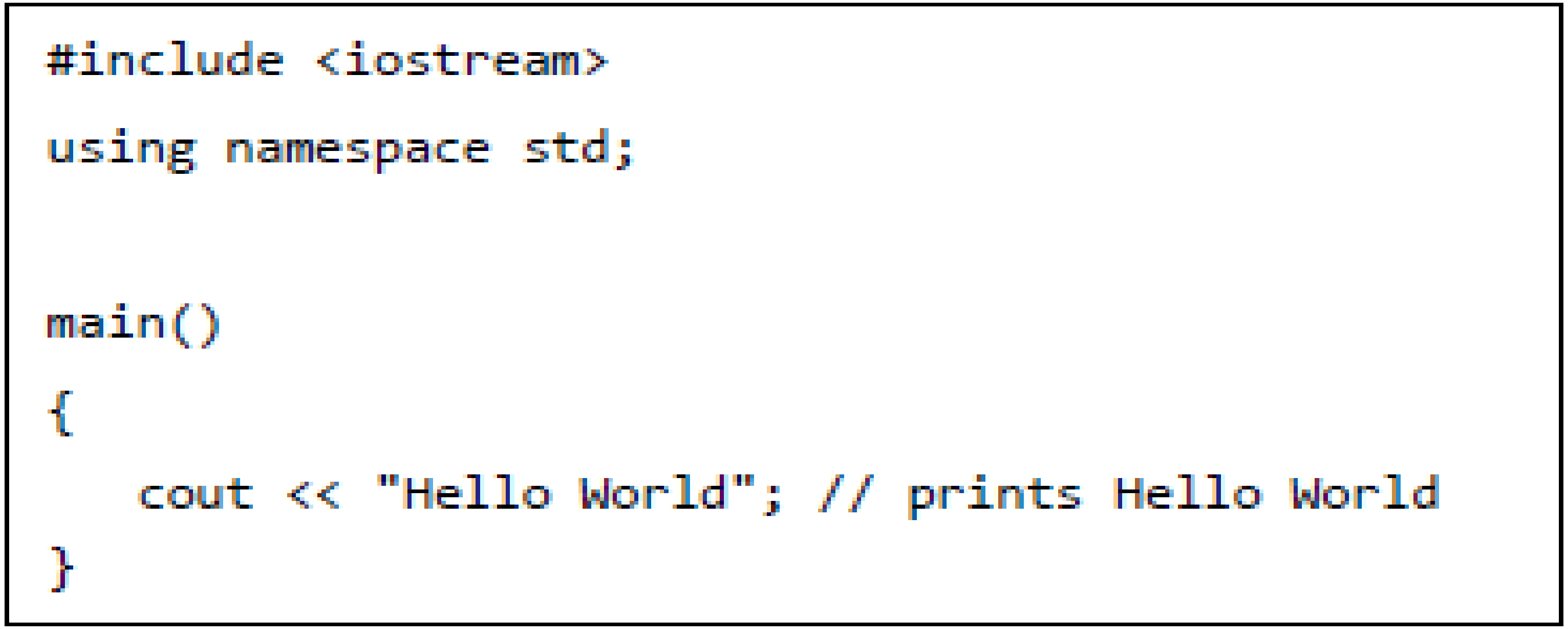
Comments in C++

- Program comments are explanatory statements that you can include in the C++ code. These comments help anyone reading the source code. All programming languages allow for some form of comments.

- C++ supports single-line and multi-line comments. All characters available inside any comment are ignored by C++ compiler.

Comments in C++

- single-line comments

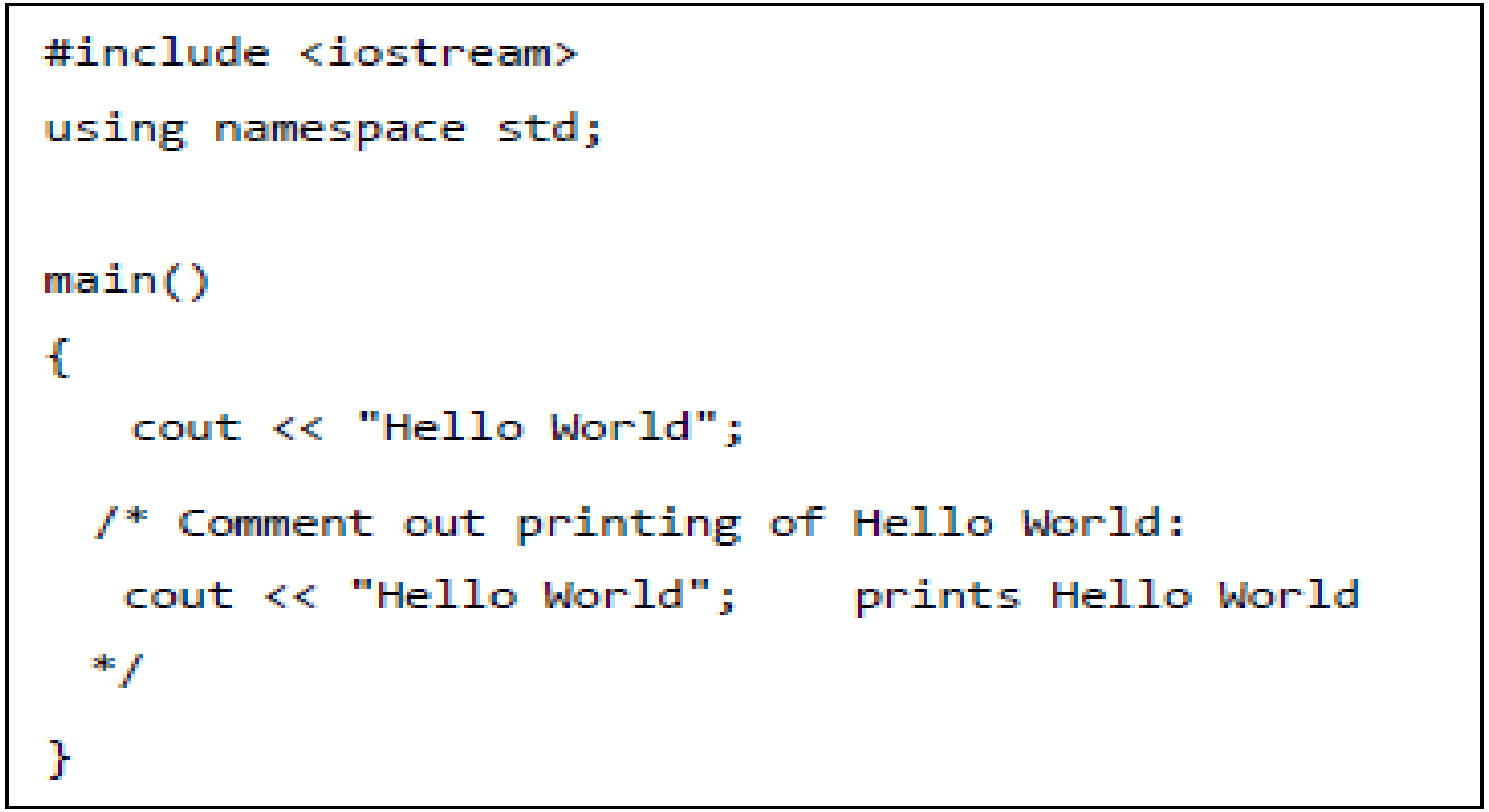


```
#include <iostream>
using namespace std;

main()
{
   cout << "Hello World"; // prints Hello World
}
```

Comments in C++

- multi-line comments



```
#include <iostream>
using namespace std;

main()
{
   cout << "Hello World";
  /* Comment out printing of Hello World:
   cout << "Hello World";    prints Hello World
  */
}
```

Questions ?

References

Online Courses YouTube playlists:	http://bit.ly/2kAPL5K
C++ Documentation	http://bit.ly/1flmcHO
CPP For School	http://bit.ly/2kifMdj
C++ Language Tutorial	http://bit.ly/1kyBMdz
C++ Language Tutorial	http://bit.ly/2rzE4hQ
C++ Tutorial Point	http://bit.ly/2BGFeO0
Fundamentals of C++ Programming	http://bit.ly/2rJHhyl
Teach Yourself C++ in 21 Days	http://bit.ly/1JXhDtL
A Complete Guide to Programming in C++	http://bit.ly/2dVkGY9