Programing Methodology in C

Lecture 2 – The Overview of C Programing language

By Elubu Joseph
MSc.IS
josebulinga@gmail.com

Agenda

1. Overview of C programing language
2. Features/Beauty of C programing language
3. Installation of IDE- Code Blocks
4. My First C program
5. Compile and Run C program

Overview of C Language

C is a structured programming language developed by Dennis Ritchie in 1973 at Bell Laboratories. It is one of the most popular computer languages today because of its structure, high-level abstraction, machine independent feature etc.

C language was developed to write the UNIX operating system, hence it is strongly associated with UNIX, which is one of the most popular network operating system in use today and heart of internet data superhighway.

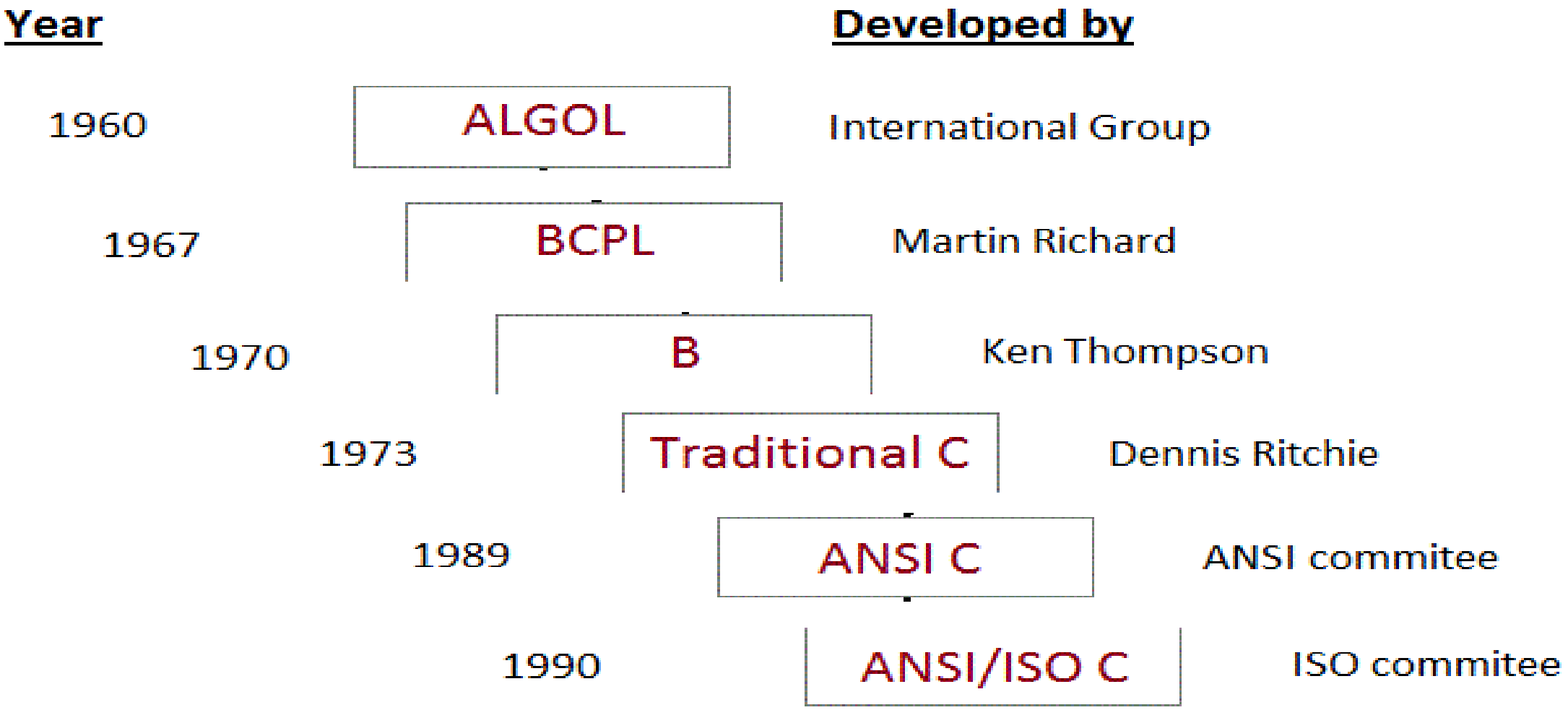
History of C language

C language has evolved from three different structured language ALGOL, BCPL and B Language. It uses many concepts from these languages while introduced many new concepts such as datatypes, structures, pointer etc. ‘

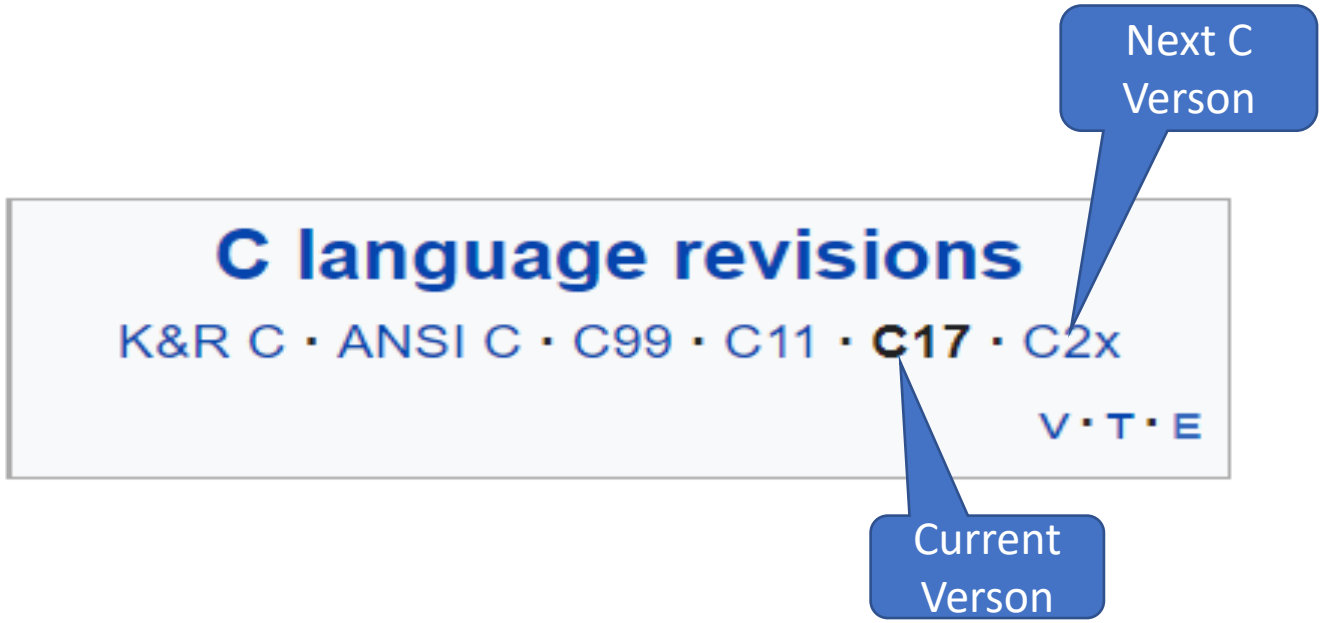
In 1988, the language was formalized by **American National Standard Institute**(ANSI).

In 1990, a version of C language was approved by the **International Standard Organization**(ISO) and that version of C is also referred to as C89.

The Geneology of C Programing language



Latest Version of C



The latest version of C language is C17, also commonly referred to as C18. The most recent standard for the C programming language, prepared in 2017 and published in June 2018, it replaced C11 and C17 will be superseded by C2x . It is supported by all the standard C language compilers.

Why C language?

The idea behind creating C language was to create an easy language which requires a simple compiler and enables programmers to efficiently interact with the machine, just like machine instructions.

C language compiler converts the readable C language program into machine instruction-Binary digits (0s and 1s)

Why is C Language so popular?

C language is a very good language to introduce yourself to the programming world, as it is a simple procedural language which is capable of doing wonders.

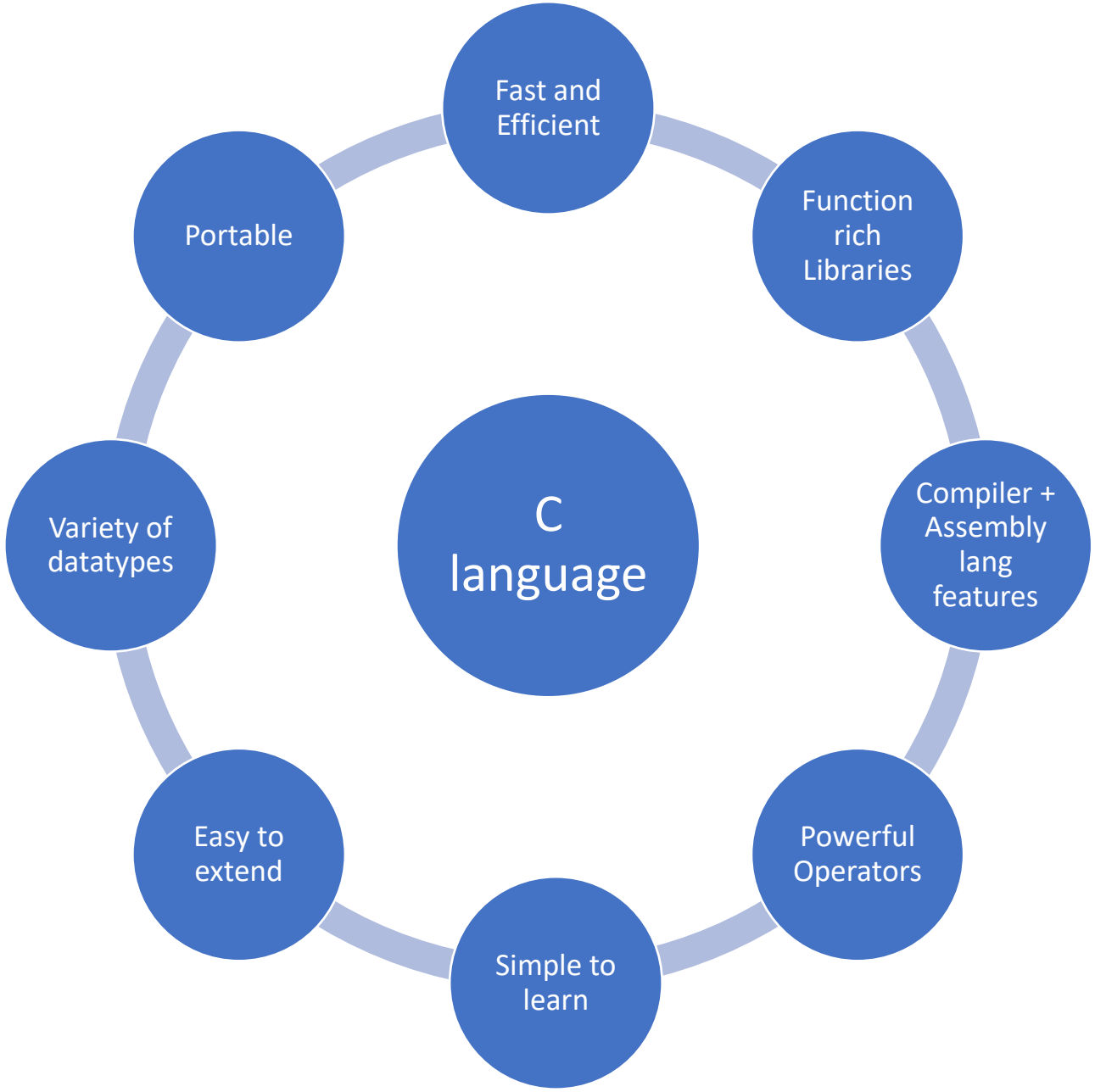
Programs written in C language takes very less time to execute and almost executes at the speed of assembly language instructions.

Why is C Language so popular? +

Widened Scope of ability;

Initially C language was mainly used for writing system level programs, like designing operating systems, but there are other applications as well which can be very well designed and developed using C language, like;- Text Editors, (interface between the hardware and user applications) like compiler, device driver, BIOS(Basic input/output system), linker, assembler etc.

Features/Beauty of C language



Features/Beauty of C language+

1. Function rich Libraries

It is a robust language with rich set of built-in functions and operators that can be used to write any complex program.

2. Compiler + Assembly lang features

The C compiler combines the capabilities of an assembly language with features of a high-level language.

Features/Beauty of C language++

3. Fast and Efficient

Programs Written in C are efficient and fast. Due to its variety of data type and powerful operators.

4. Variety of datatypes

The basic data types are integer-based and floating-point based. C language supports both signed and unsigned literals.

Types	Data Types
Basic Data Type	int, char, float, double
Derived Data Type	array, pointer, structure, union
Enumeration Data Type	enum
Void Data Type	void

Features/Beauty of C language +++

6. Easy to extend

Another important feature of C program, is its ability to extend itself. According to Christensson, P. (2014), If a programming language is extensible, it may support custom syntax and operations

This means that C supports custom syntax and operations. These custom elements can be defined in the source code and are recognized by the compiler along with the pre-defined elements.

Features/Beauty of C language ++++

7. Simple to learn

C Language is an amazing language when it comes to simplicity of syntax with decent functionality.

It is a perfect mix of both the above features, which makes it the best contender to be taught to students who have just started learning coding, to introduce them into the programming world.

Features/Beauty of C language +++++

8. Portable

If you write a C code in your machine, it will run on any machine which supports C, without modifying a single line of code. **Because it is not tied to any hardware or system,** (log2base2.com n.d).

Features/Beauty of C language +++++

9. Powerful Operators

C operators can be classified into the following types(Studytonight.com)

- Arithmetic operators
- Relational operators
- Logical operators
- Bitwise operators
- Assignment operators
- Conditional operators
- Special operators

Write, Compile and Run C Program

- To write, compile and run a C language program, you need a C compiler and other tools necessary to handle your code. To setup a C language compiler in your Computer/laptop, there are two ways:
- Download and install a fully fledged IDE like Code::Blocks, Turbo C or Microsoft Visual C++, which comes along with a C language compiler.
- Or, you use any text editor to edit the program files and download the C compiler separately.

NOTE? We will base our studies on Code::blocks IDE in this course. So you need to download codeblocks-20.03mingw-setup.exe or later version. **Note!.** *Consider a download option with* ***mingw*** *in its name.*

Download and Installation of an IDE

Before we install IDE- Code Blocks into our computer(s), Lets Download one “codeblocks-20.03mingw-setup.exe” in our case, from Sourceforge.net by clicking or pasting the link below on you web address bar.

https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe/download

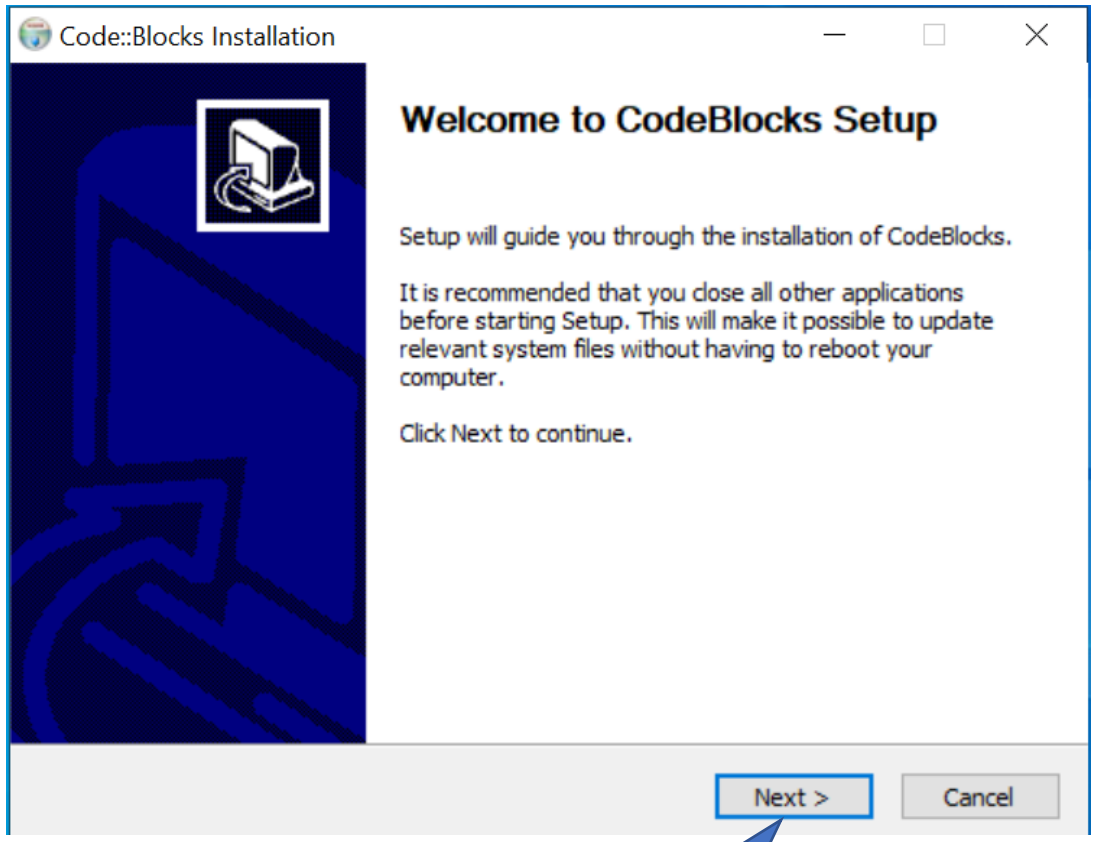
Installation of an IDE

Having download our IDE, we now need to install it into our computer.

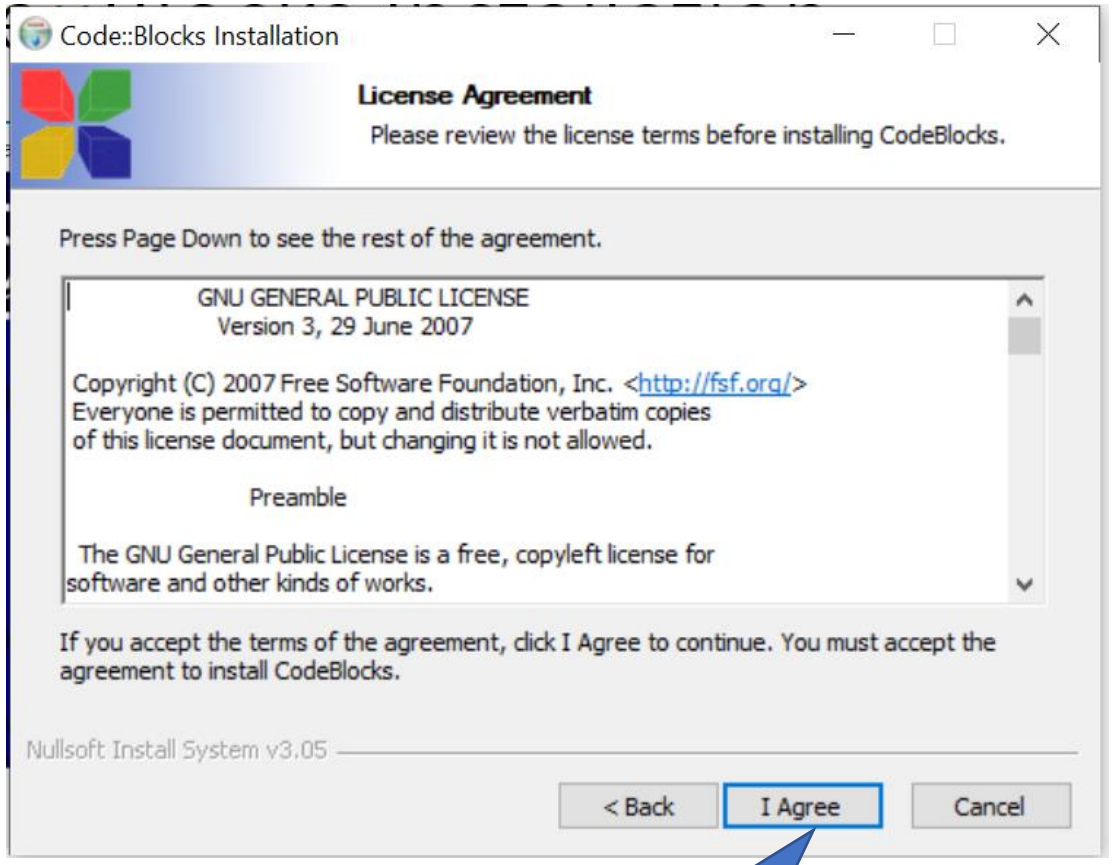
How?

- Open the folder where your download is
- Double click on the codeblocks-20.03mingw-setup.exe file to open
- Follow the steps as guided by the installation wizard

Code::Blocks IDE Installation

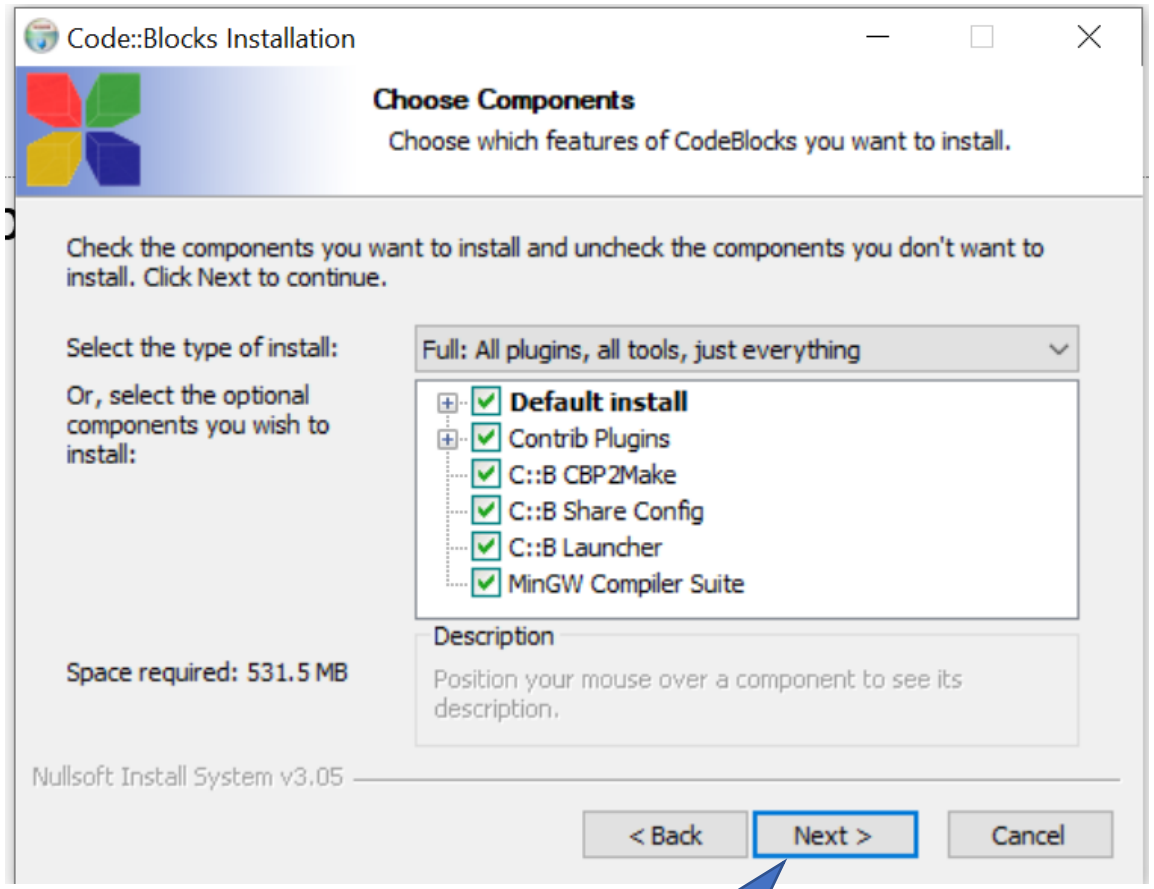


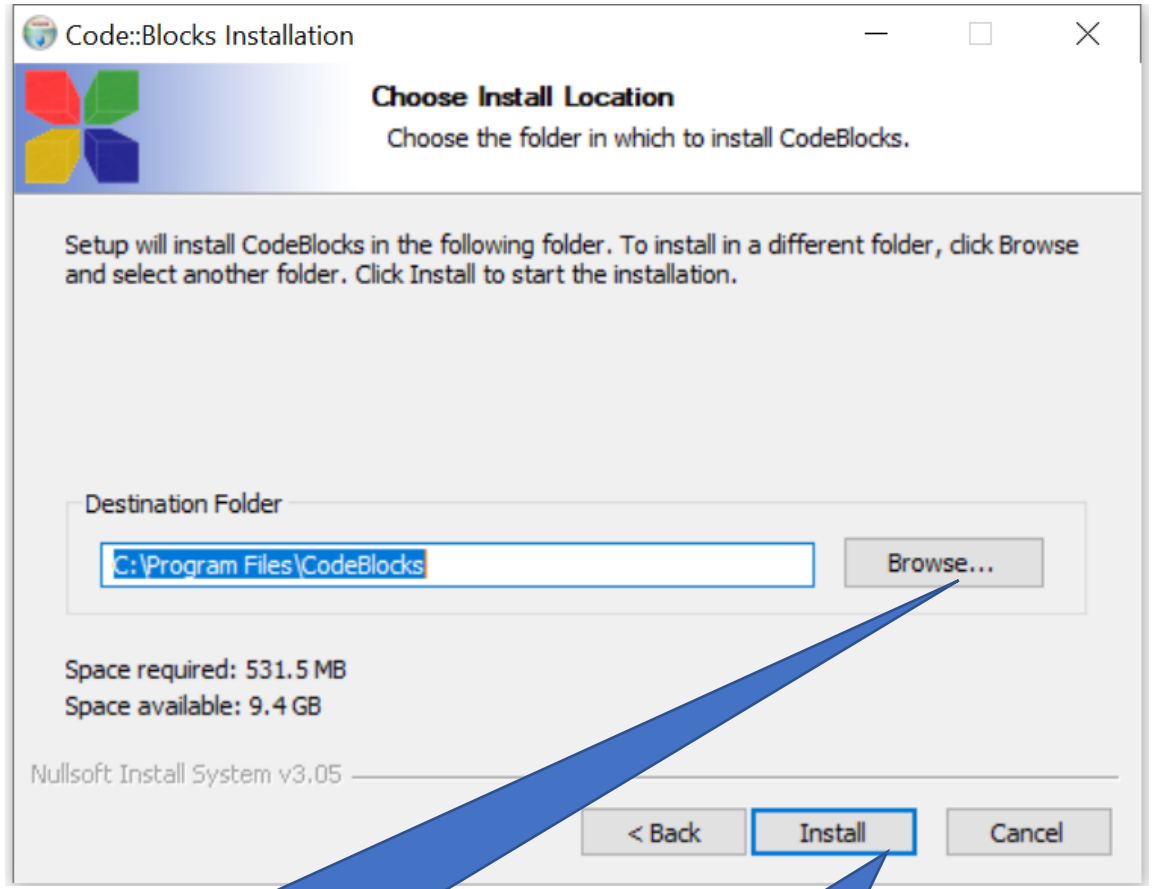
Click Next



Click I agree

Code::Blocks IDE Installation



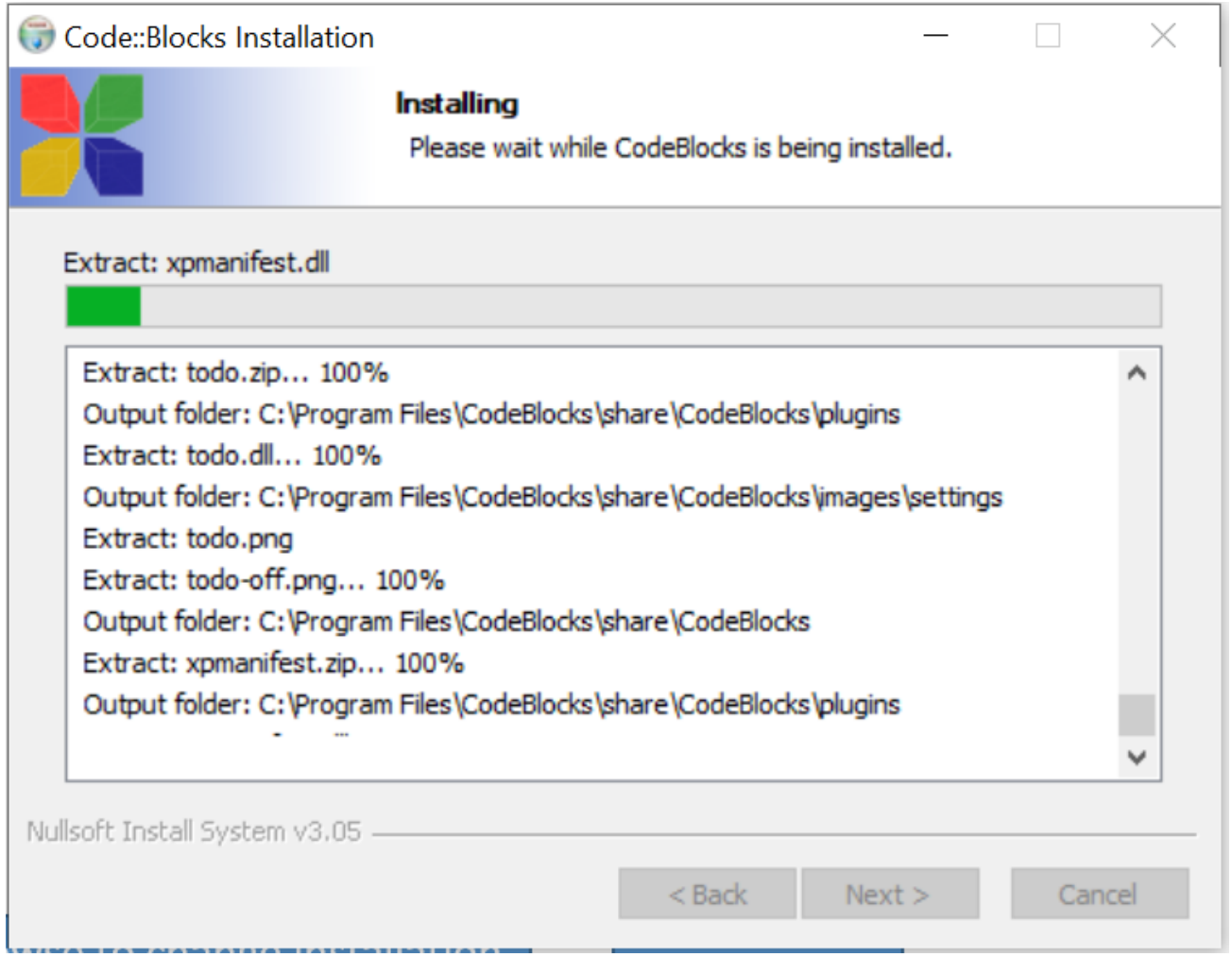


Click Next

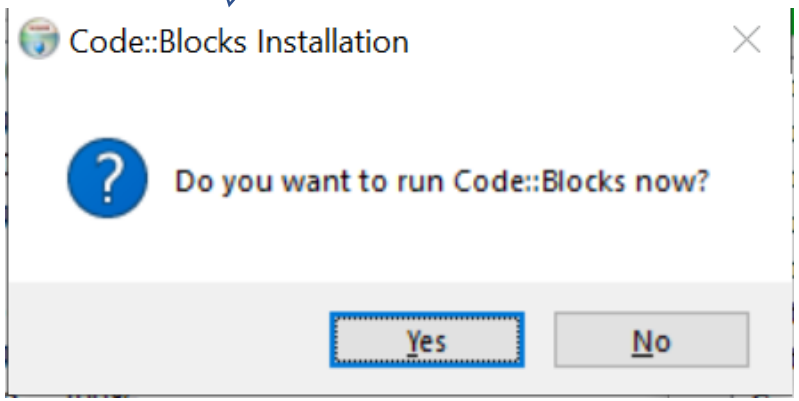
Click Browse to change installation directory if necessary

Click Install

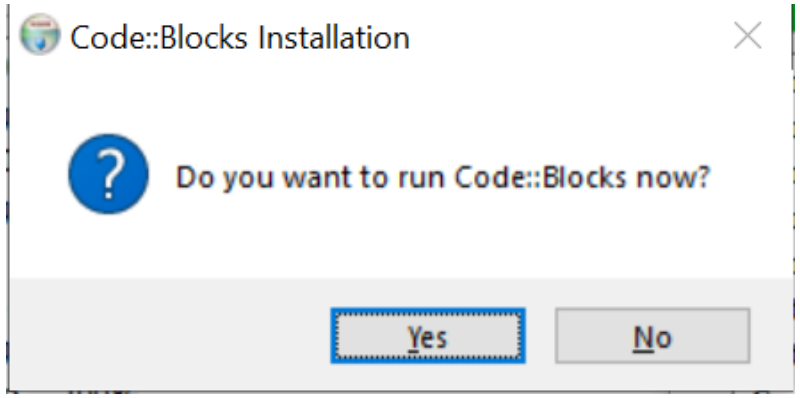
Code::Blocks IDE Installation

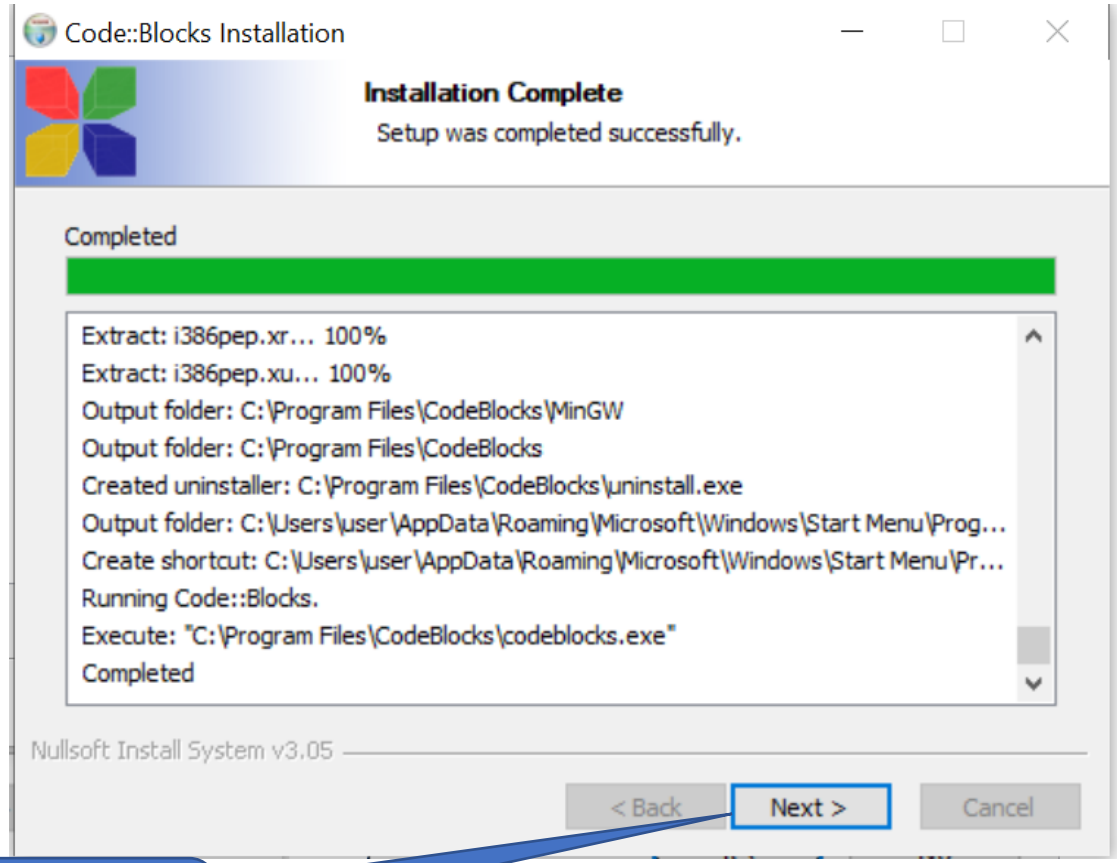


Installation in progress Just relax and wait until you See the dialog below asking you whether you want to run Code::Blocks now.



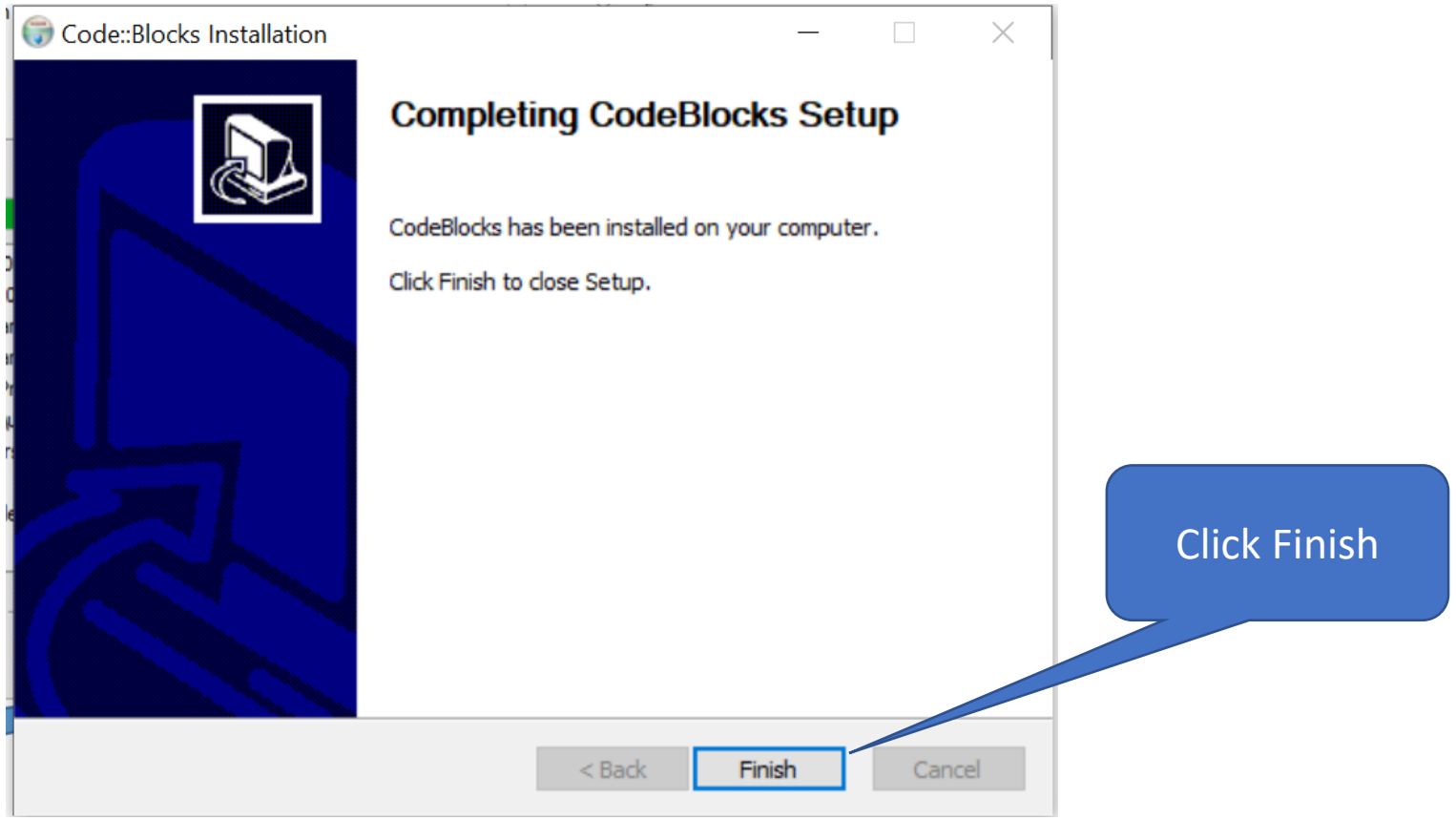
Code::Blocks IDE Installation Completion





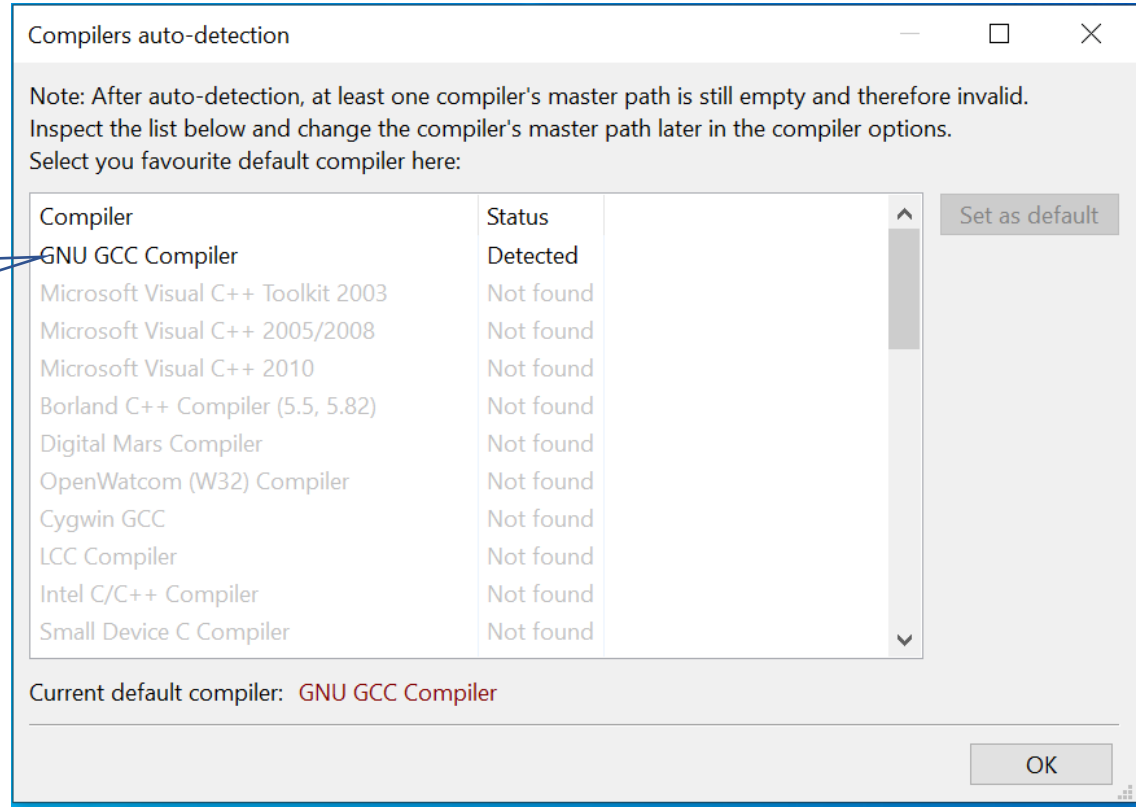
After clicking Yes or No in the previous dialog box, you can now **Click Next** to complete the installation process

Code::Blocks IDE Installation Completion



Compiler Auto-Detection Setup

When you open Code::Blocks for the first time, it detects the Compiler **GNU GCC Compiler** automatically.

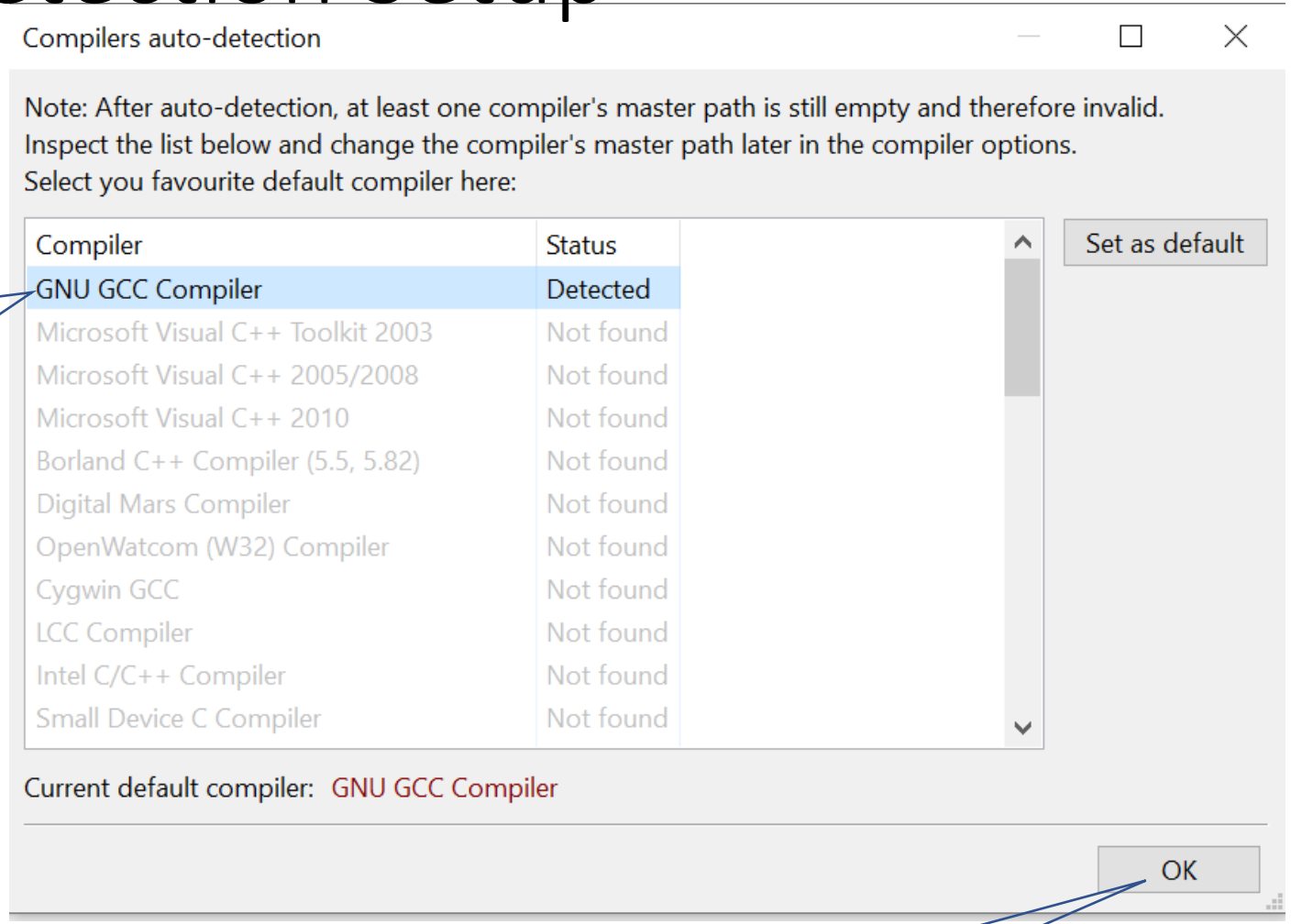


Compiler Auto-Detection Setup+

Click on GNU GCC Compiler

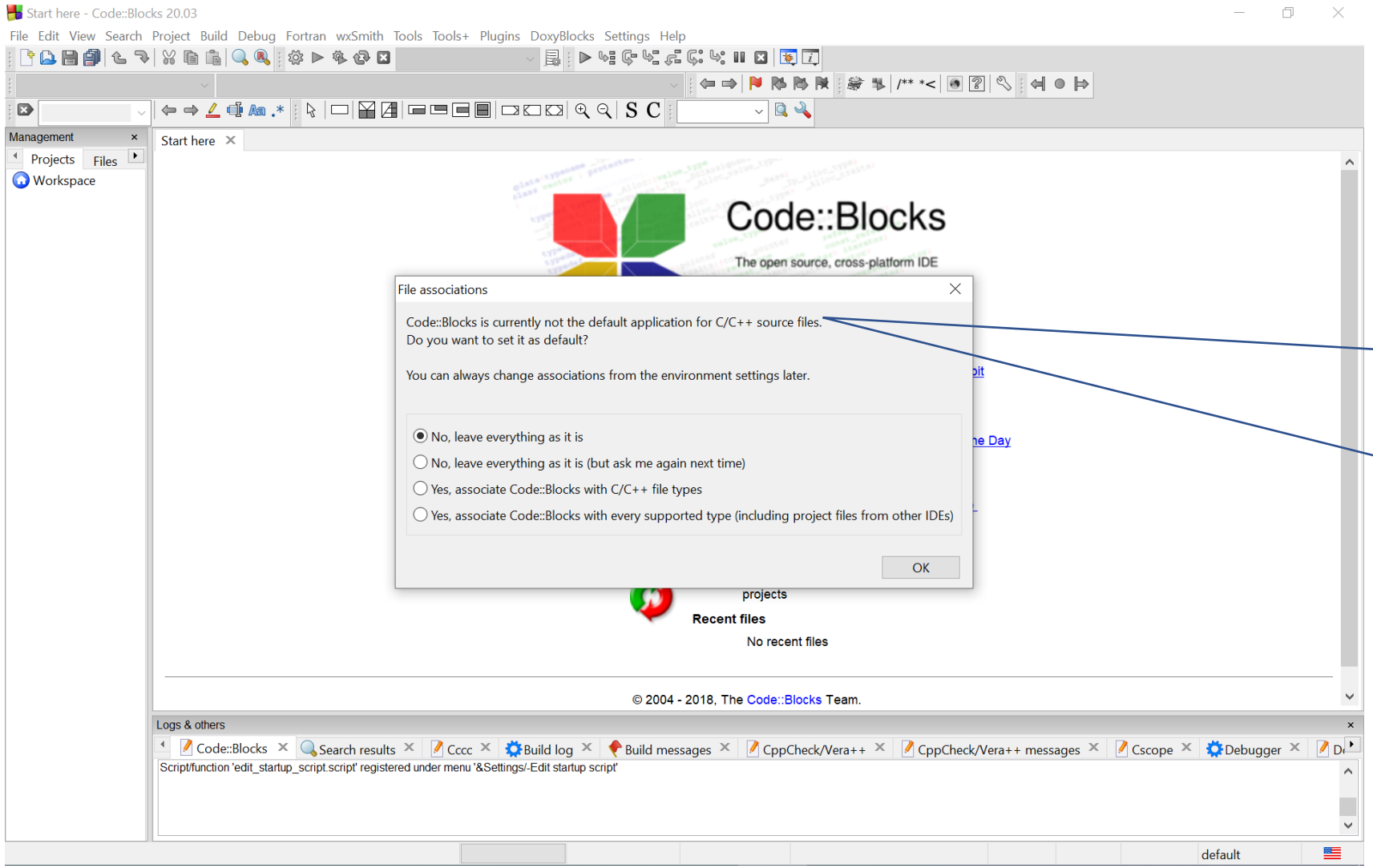
Lebel

Then Click OK



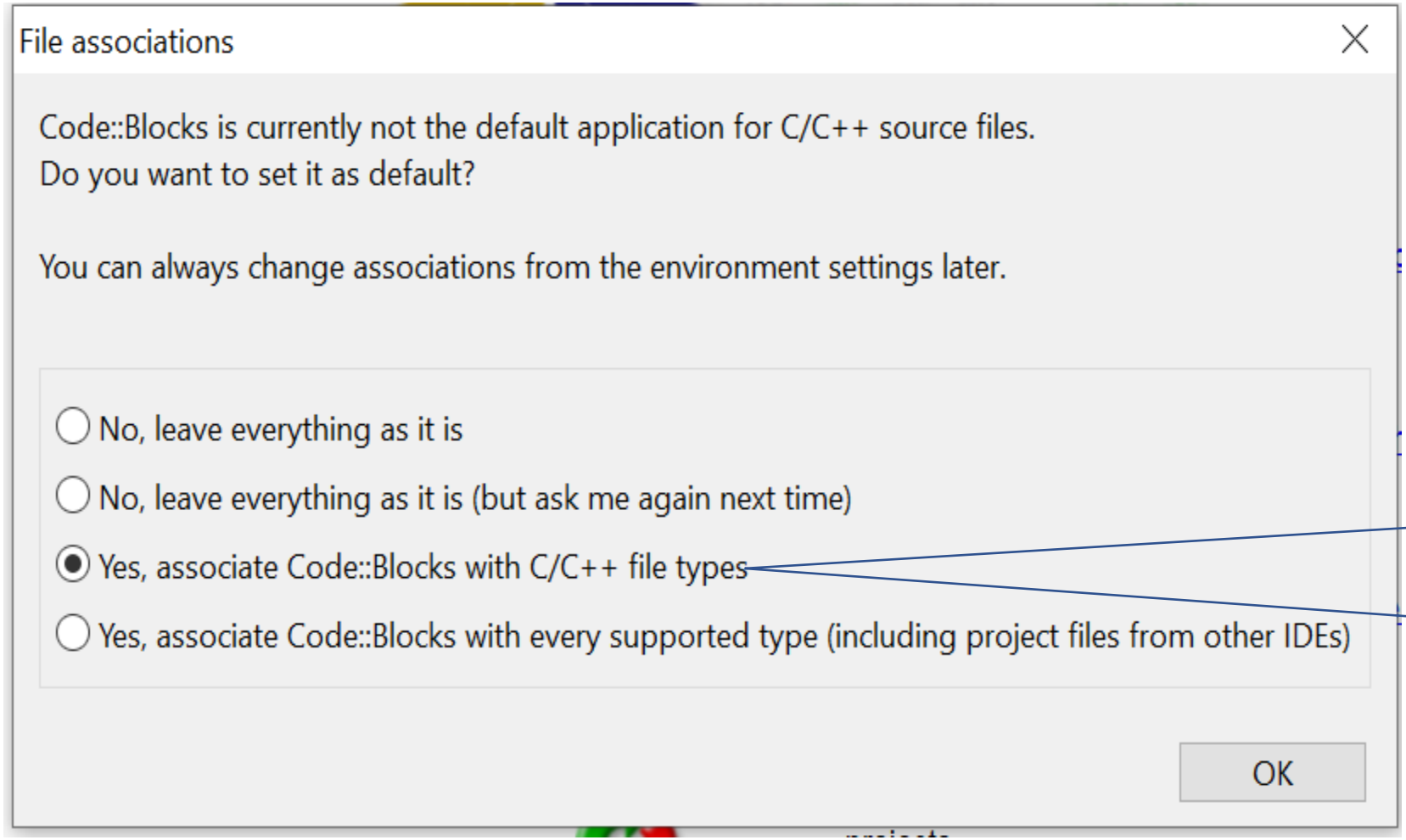
Click OK

Setting Code::Blocks to be Default C/C++ Application



Code::Blocks is currently not the default application for C/C++ Source Files Do you want to set it as default?

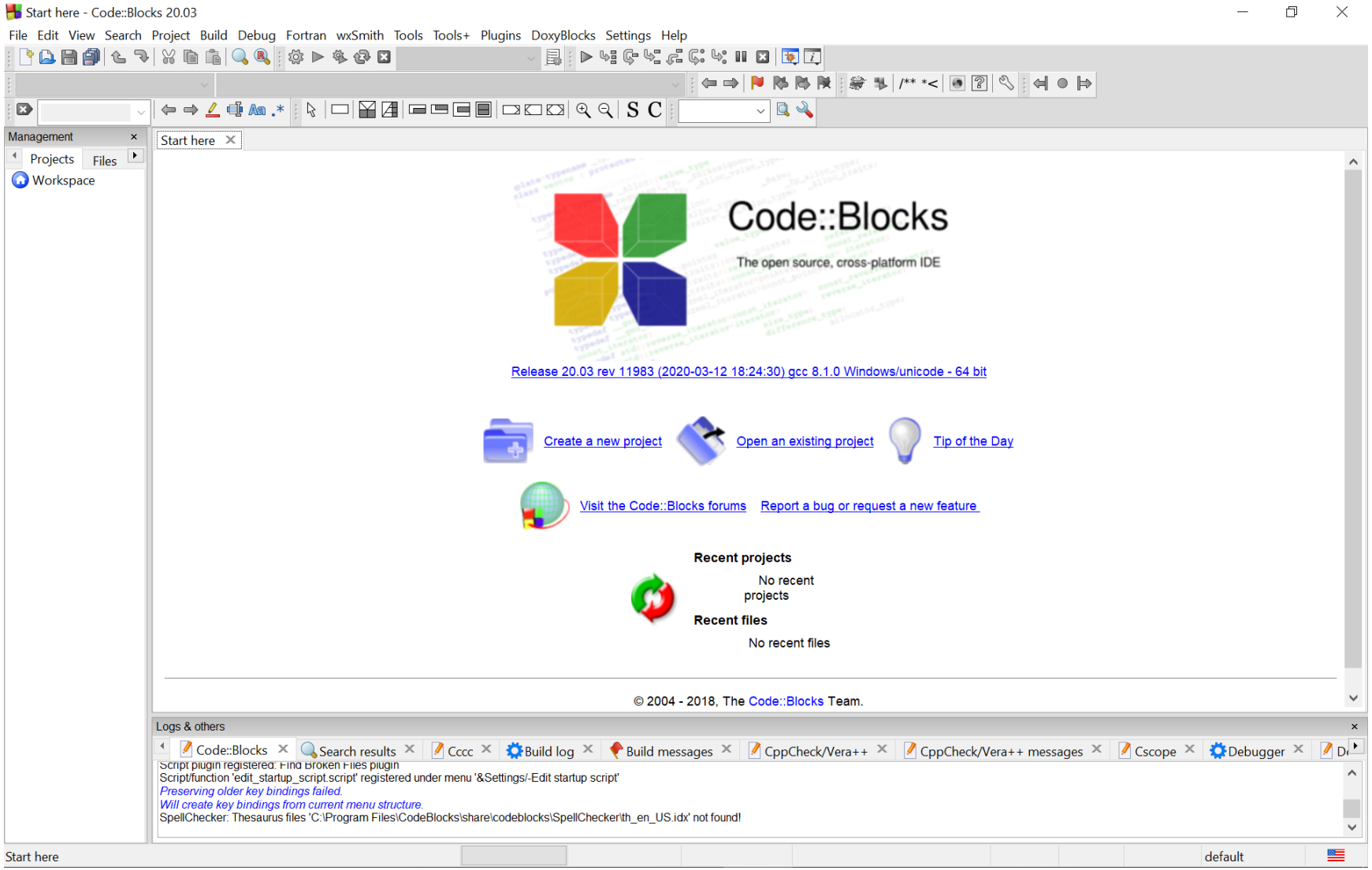
Setting Code::Blocks to be Default C/C++ Application



Click Yes, associate Code::Blocks with C/C++ file types Radio button

Then **Click OK** button

Congratulations



By the fact that you're now seeing the above code blocks display, you are done with installation and setting up your IDE

Create Source File Directory

Before we can actually start creating our programs, we need to create source files directory.

In this section, we will move step by step through the process of creating a source file directory, source file, entering, compiling, and running a simple program.

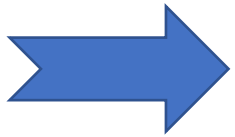
Creating a source file directory

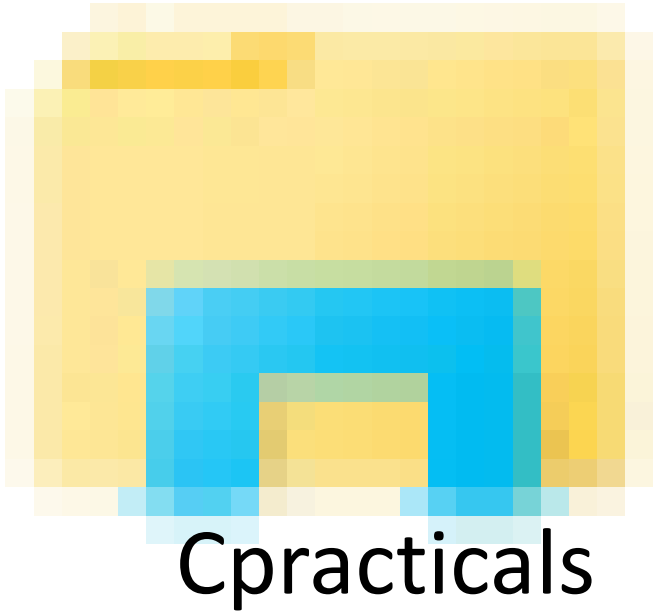
Before we enter into things, lets first create a folder where our files will be stored.

Create a folder on the Desktop of your computer, call it **Cpracticals.**

How?

i. **Right click** on the Desktop
ii. **Point** at New
iii. Click Folder
iv. **Type** the name “Cpracticals”
v. Press **Enter Key** to Confirm the name
vi. Finally Click **any empty Space** on the desktop to deselect the created folder.





Create the Program

We can now create our first program called “Hello Planate.c”, But lets first Open our IDE (**Code::Blocks**)

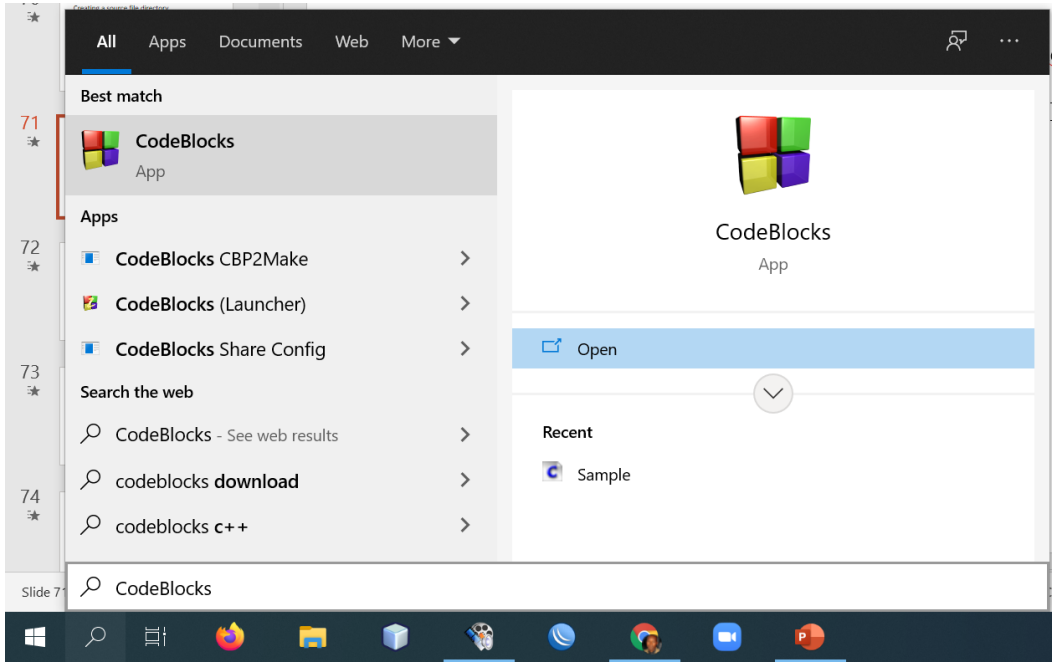
How?

i. Click on Windows Search icon/box

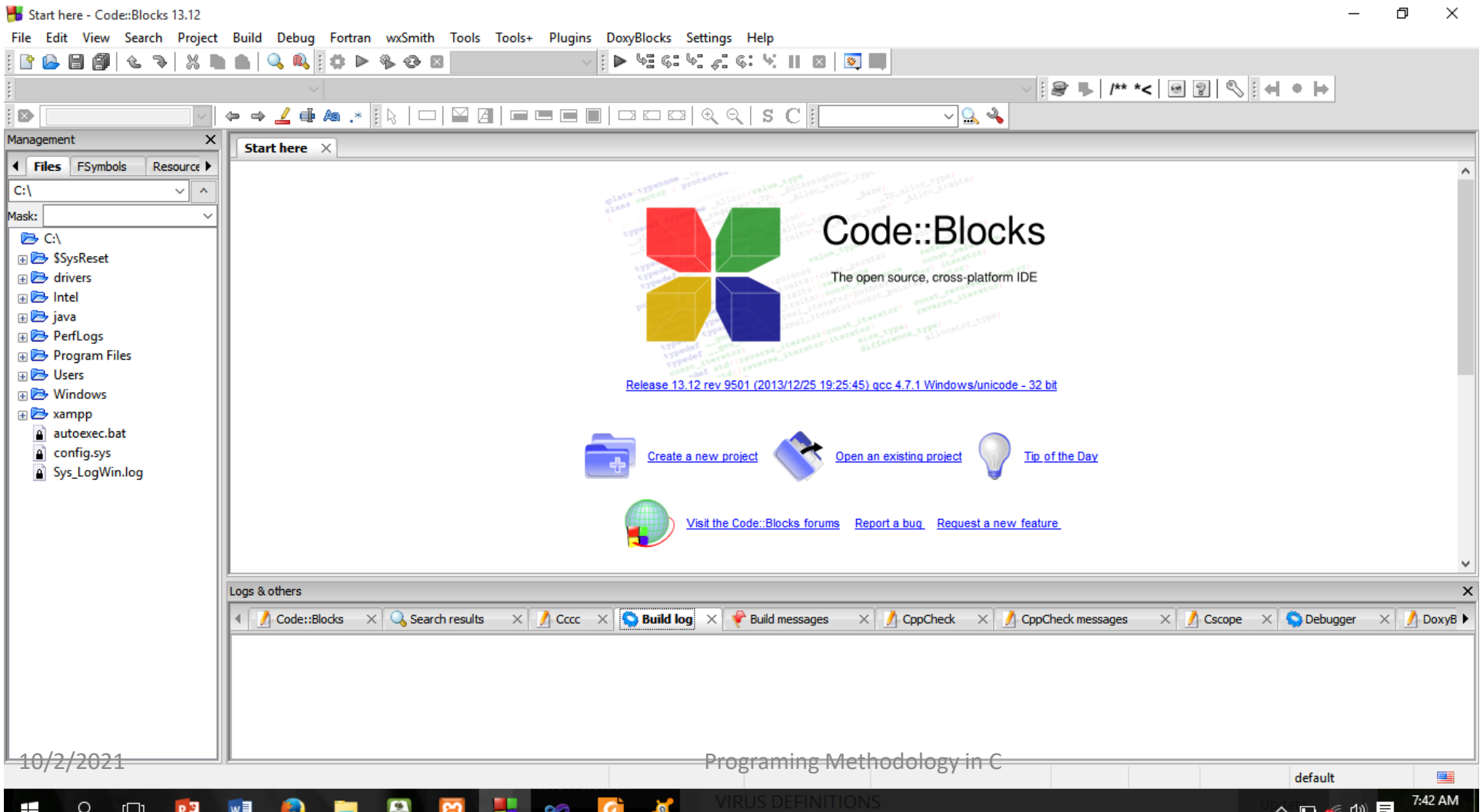
ii. Enter Code Blocks

iii. Click on Code Blocks

See



You should be seeing the picture below after opening Code::Blocks



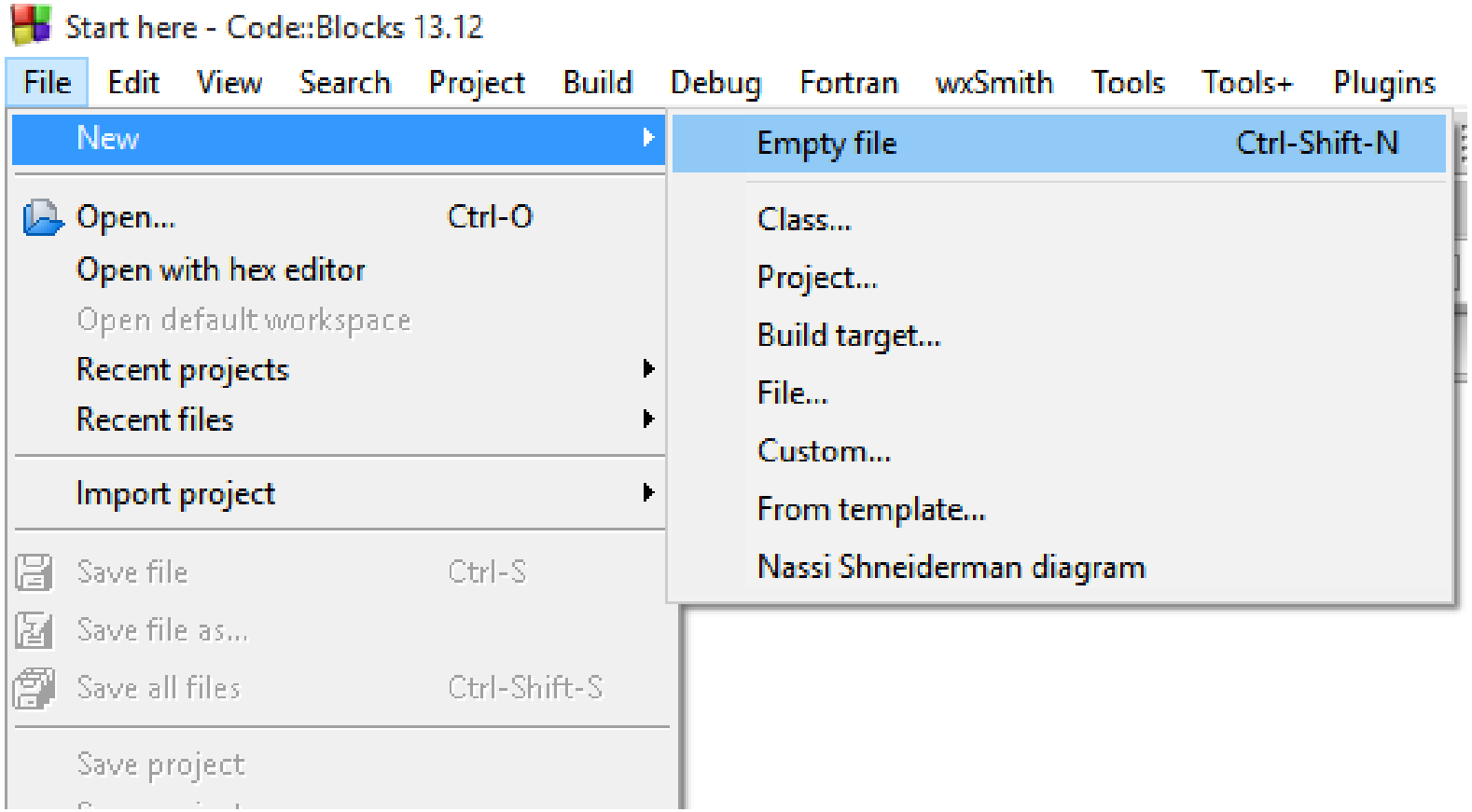
Create and save your source File now

How?

Click File menu

Point at New

Click Empty File



Save your File as HelloPlanate.c

How?

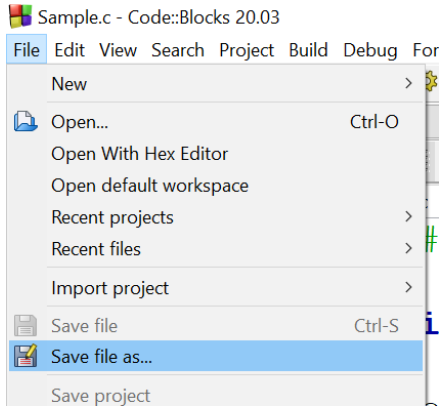
i. Click on File Menu

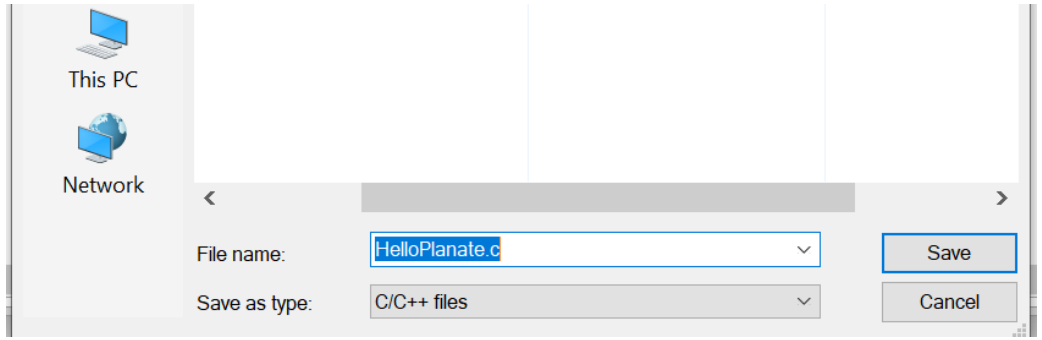
ii. Click **Save file as…**

iii. Navigate to your Folder called “Cpraticals” found on the Desktop

iv. Type the name **“HelloPlanate.c”** of the file in the File Name text box

v. Click **Save** Button





First C Program Source Code

Lets see how to write a simple and most basic C program:

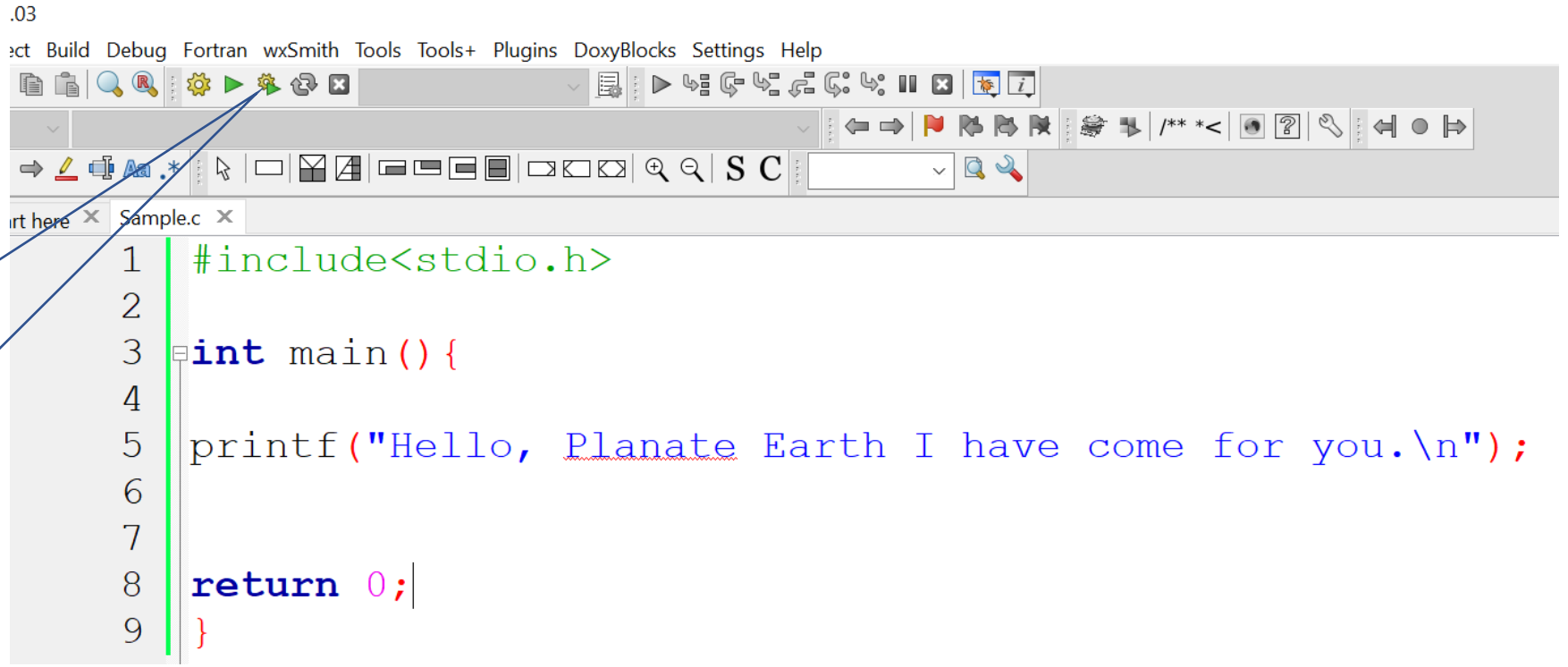
```c
#include <stdio.h>
int main() {
   printf(" Hello Planate Earth, I have come for you\n!");
   //single line comment

   return 0; /* multi line comments */
}
```

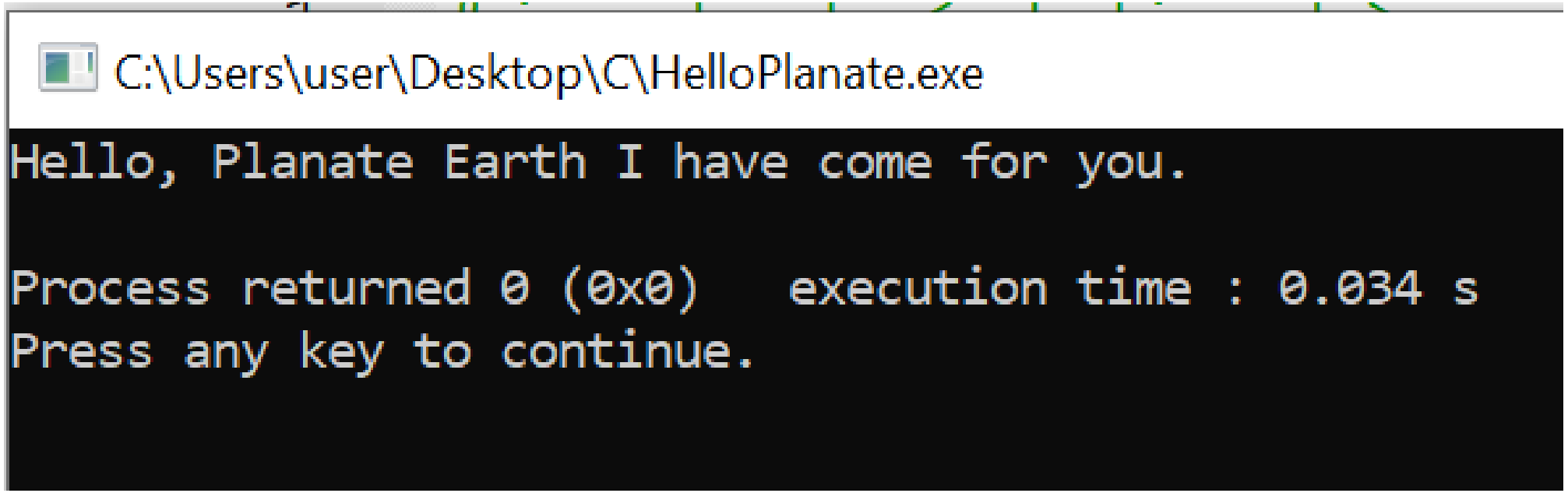
OUTPUT: Hello Planate Earth, I have come for you!

Build and Run the Code

Click on **Build and Run icon**



Output


C:\Users\user\Desktop\C\HelloPlanate.exe
Hello, Planate Earth I have come for you.
Process returned 0 (0x0)   execution time : 0.034 s
Press any key to continue.

Analysis of the life of a program from Conception, Coding to Execution

Where dose Programs Start?

Programs start out as an idea in a programmer's head/Mind. He uses a text editor e.g. *code blocks* to write his thoughts into a file called a *source file*, containing *source code.*

This file is transformed by the ***compiler*** into Assembly code which the ***assembler*** then transform to an ***object file***. Next, a program called the ***linker*** takes the object file, combines it with predefined routines from a ***standard library***, and produces an *executable program* (a set of machine-language instructions), (Steve O, 1997).

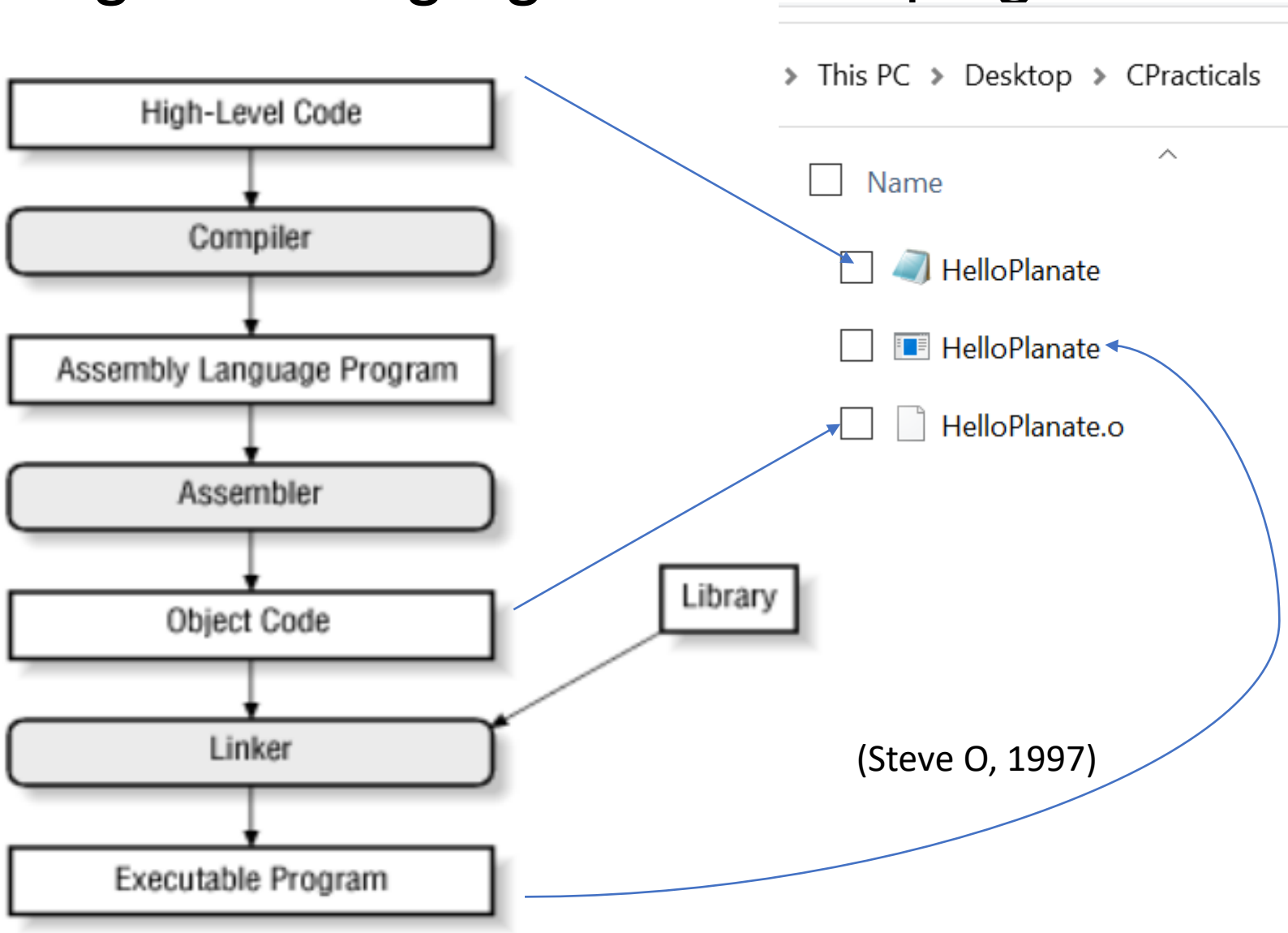
Analysis of the life of a program from Conception, Coding to Execution +

C programs codes are written in a high-level language using letters, numbers, and the other symbols you find on a computer keyboard.

Computers actually execute a very low-level language called *machine code* (a series of numbers e.g. 11001001 or 10101011- bits). So, before a program level can be used, it must undergo several transformations.

In the following sections, we'll see how these various forms of the IDE program components work together to produce the final program. **See the figure below**

Transformation of a high-level language code into a program



How does the IDE know which tools to use during code transformation to a program

“A wrapper" is a program that determine which tools needed to be run and then run them. You don't have to run the compiler, assembler, and linker individually. Most C compilers use a wrapper.

Some programming systems go even further and provide the developer with an Integrated Development Environment (IDE).

The IDE contains an editor, compiler, linker, project manager, debugger, and more in one convenient package. E.g of IDEs Code Blocks, Borland etc.

Summary

In summary therefore, we looked at:-

- Overview of C programing language

  Genealogy of C,

- Features/Beauty of C programing language

  Function rich libraries, Portability, Efficient etc.

- Installation of IDE- Code Blocks

  Downloaded, install and configured Code::Blocks IDE

- My First C program

  Created:- source file directory, source file, wrote code

- Compile and Run C program

  Used Build and Run icon to compile and ran our program.

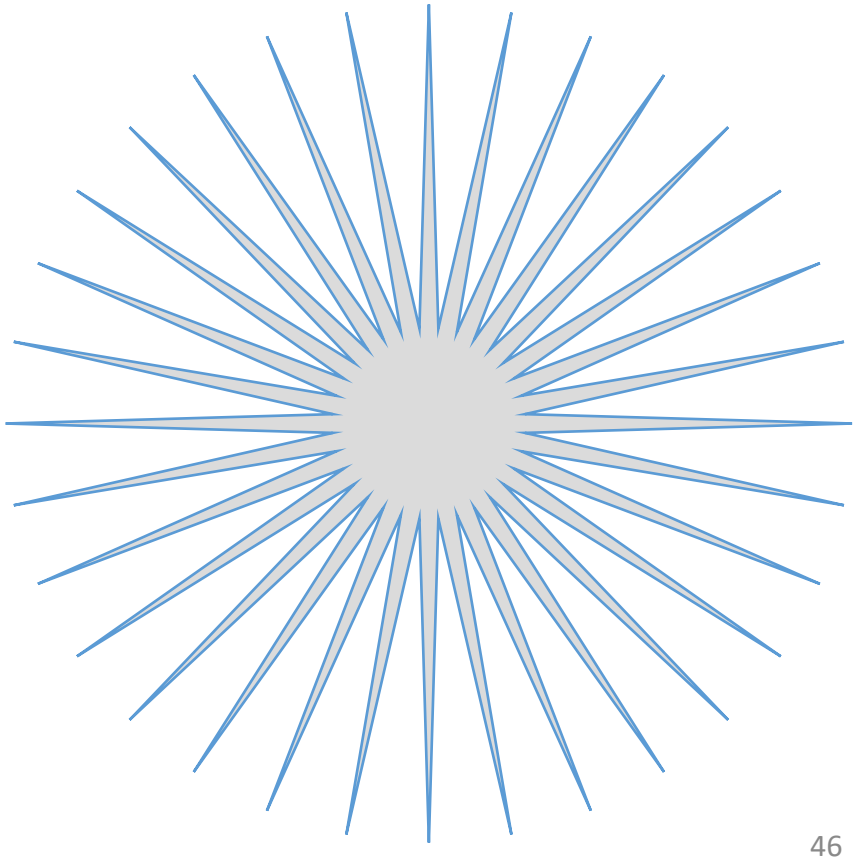
Review Questions

1. Explain how a wrapper works?
2. State any four components of a fully fledged IDE.
3. Explain the geneology of C programing language to C17.
4. Mention the compiler we are using in our Code::Blocks IDE.

Next Lecture

- Writing a program in C
- C Program Structure

Thank you for you attention



References

Christensson, P. (2014, October 31). *Extensible Definition*. Retrieved 2021, Sep 30, from https://techterms.com

Introduction to C language. (n.d.). Retrieved September 30, 2021, from https://www.log2base2.com/C/basic/introduction-to-c-language.html.