BCSE102L- Structured and object-oriented programming

Module 1

Dr. P.Keerthika
Associate Professor
School of Computer Science & Engineering
VIT, Vellore.

BCSE102L- Structured and object-oriented programming – Course Objective

1. To impart the basic constructs in structured programming and object oriented paradigms.
2. To inculcate the insights and benefits in accessing memory locations by implementing the real world problems.
3. To help solving the real world problems through appropriate programming paradigms.

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

BCSE102L- Structured and object-oriented programming – Course outcome

1. Understand different programming language constructs and decision making statements, manipulate data as a group.
2. Recognize the application of modular programming approach, create user defined data types and idealize the role of pointers.
3. Comprehend various elements of object oriented programming paradigm, propose solutions through inheritance and polymorphism, identify the appropriate data structure for the given problem and devise solution using generic programming techniques.

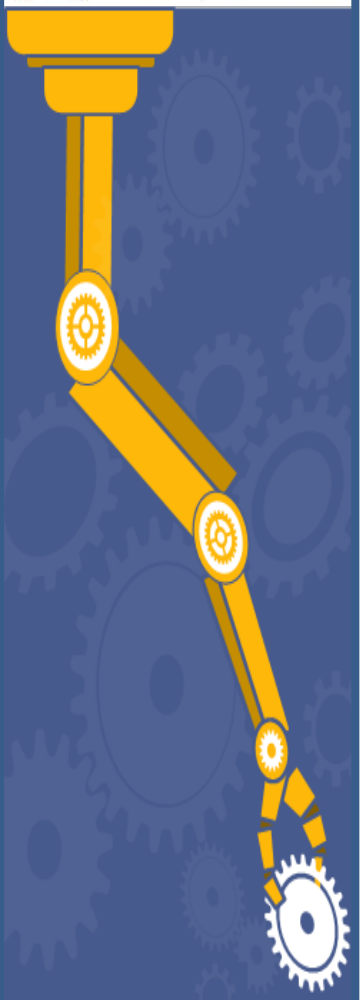
BCSE102L- Structured and object-oriented programming

Module:1	**C Programming Fundamentals**	**2 hours**
Variables - Reserved words – Data Types – Operators – Operator Precedence - Expressions - Type Conversions - I/O statements - Branching and Looping: if, if-else, nested if, if-else ladder, switch statement, goto statement - Loops: for, while and do…while – break and continue statements.		
Module:2	**Arrays and Functions**	**4 hours**
Arrays: One Dimensional array - Two-Dimensional Array – Strings and its operations. User Defined Functions: Declaration – Definition – call by value and call by reference - Types of Functions - Recursive functions - Storage Classes - Scope, Visibility and Lifetime of Variables.		
Module:3	**Pointers**	**4 hours**
Declaration and Access of Pointer Variables, Pointer arithmetic – Dynamic memory allocation – Pointers and arrays - Pointers and functions.		
Module:4	**Structure and Union**	**2 hours**
Declaration, Initialization, Access of Structure Variables - Arrays of Structure - Arrays within Structure - Structure within Structures - Structures and Functions – Pointers to Structure -		
Module:5	**Overview of Object-Oriented Programming**	**5 hours**
Features of OOP - Classes and Objects - "this" pointer - Constructors and Destructors - Static Data Members, Static Member Functions and Objects - Inline Functions – Call by reference - Functions with default Arguments - Functions with Objects as Arguments - Friend Functions and Friend Classes.		
Module:6	**Inheritance**	**5 hours**
Inheritance - Types of Inheritance: Single inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance - Multipath Inheritance - Inheritance and constructors.		
Module:7	**Polymorphism**	**4 hours**
Function Overloading - Operator Overloading – Dynamic Polymorphism - Virtual Functions - Pure virtual Functions - Abstract Classes.		
Module:8	**Generic Programming**	**4 hours**
Function templates and class templates, Standard Template Library.		
	Total Lecture hours:	**30 hours**

BCSE102L- Structured and object-oriented programming – Text Books and Reference Books

Text Book(s)	
1.	Herbert Schildt, C: The Complete Reference, 4th Edition, McGraw Hill Education, 2017
2.	Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw Hill Education, 2017.
Reference Books	
1.	Yashavant Kanetkar, Let Us C: 17th Edition, BPB Publicaitons, 2020.
2.	Stanley Lippman and Josee Lajoie, C++ Primer, 5th Edition, Addison-Wesley publishers, 2012.

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)



BCSE102L- Structured and object-oriented programming

Assessment Configuration(THEORY)

Assessment method	Syllabus Coverage	Date	Marks
QUIZ-1	Unit 1,2,3	**Before CAT1**	**10**
CAT-1	As decided in Course Committee Meeting	As Scheduled by Dean Academics	**15(50)**
QUIZ-2	Unit 4,5,6	**Before CAT 2**	**10**
CAT-2	As decided in Course Committee Meeting	As Scheduled by Dean Academics	**15(50)**
Digital Assignment	All Modules	**Before FAT**	**10**
FAT	All Modules	As Scheduled by Dean Academics	**40(100)**

BCSE102P- Structured and object-oriented programming Laboratory

Indicative Experiments	
1.	Programs using basic control structures, branching and looping
2.	Experiment the use of 1-D, 2-D arrays and strings and Functions
3.	Demonstrate the application of pointers
4.	Experiment structures and unions
5.	Programs on basic Object-Oriented Programming constructs.
6.	Demonstrate various categories of inheritance
7.	Program to apply kinds of polymorphism.
8.	Develop generic templates and Standard Template Libraries.

Text Book(s)	
1.	Robert C. Seacord, Effective C: An Introduction to Professional C Programming, 1st Edition, No Starch Press, 2020.
Reference Book(s)	
1.	Vardan Grigoryan and Shunguang Wu, Expert C++: Become a proficient programmer by learning coding best practices with C++17 and C++20's latest features, 1st Edition, Packt Publishing Limited, 2020.

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

BCSE102L- Structured and object-oriented programming

Assessment Configuration(LAB)

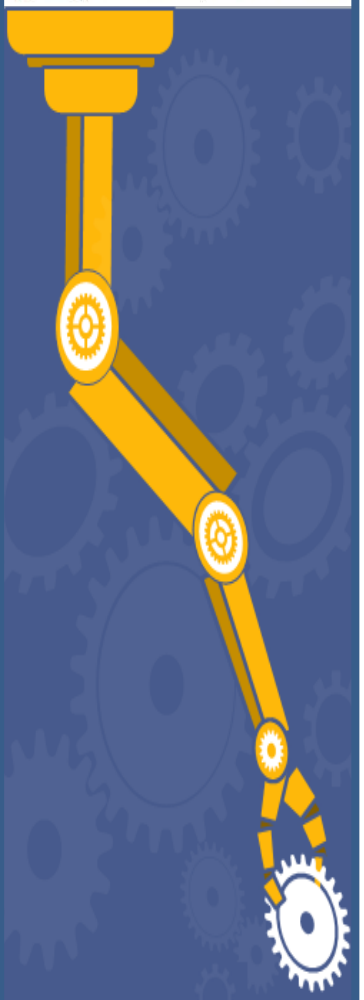
Assessment Component	Syllabus Coverage	Marks	Pattern
Problem set-I	Module 1 & 2	10	Take home assignment
Problem set-II	Module 3 & 4	10	Take home assignment
Problem set-III	Module 5 & 6	10	Take home assignment
Problem set-IV	Module 7 & 8	10	Take home assignment
Lab CAT	All the 8 Modules	20	To be conducted with one Problem each in C and C++
FAT	All the 8 Modules	40	To be conducted for 50 marks with one Problem each in C and C++

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Lab Component Assessment

- Assessment questions will be uploaded in VTOP/ intimated in class.
- An assessment will include questions from multiple lab sessions
- Programming type (C,C++)
- Attach the screenshot of the output after the code.
- Check the output for multiple and diverse test cases and include all screenshots.
- Only a single PDF document is allowed per assessment.
- Submission can be done only through VTOP

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)



General Guidelines

- All DAs must be submitted through VTOP only.
- No other mode (such as email) is accepted
- Submit assignments well in advance to avoid probable network issues/VTOP login errors at the last moment.

Queries????

All the Best

BCSE102L- Structured and Object-Oriented Programming - MODULES

- Module-1: C Program Fundamentals
- Module-2: Arrays and Functions
- Module-3: Pointers
- Module-4: Structure and Union
- Module-5: Overview of object Oriented Programming
- Module-6: Inheritance
- Module-7: Polymorphism
- Module-8: Generic Programming

BCSE102L- Structured and Object-Oriented Programming

- **Module-1:C Program Fundamentals - Introduction**
  - **Variables**
  - **Reserved Words**
  - **Data types**
  - **Operators- Operator Precedence**
  - **Expressions**
  - **Type Conversions**
  - **I/O Statements**

BCSE102L- Structured and Object-Oriented Programming

- **Module-1: C Program Fundamentals – Branching and Looping**
  - if, if-else, nested if, else-if ladder
  - Switch Statement
  - Goto Statement
  - Looping
    - For
    - While and do while
  - Break Statement
  - Continue Statement

BCSE102L- Structured and Object-Oriented Programming

- **What is Program?**
  - Set of instructions that instruct the CPU to perform a defined task.
  - We can write computer programs using various programming languages
  - Programming paradigm is a way of categorizing a programming language depending on its features
    - **Structured Programming**
    - **Object-Oriented Programming**

What is Structured Programming??

Structured/ Modular Programming

- Developing a program using a set of **modules or functions**

- Subset of Procedural Programming, user-friendly and easy to understand.
- Functions have statements embraced inside curly braces.
- Each Functions represents a functionality and performs a specific task.
- Main method communicates with functions by calling those functions in the main program

What is Structured Programming??

Structured/ Modular Programming

- Easier for the programmer to test and debug the code.
- Top down Approach
- Difficult to modify.
- Data is not secure in structured programming.
- Difficult to reuse code in structured programming

Why did We Learn Python?

- Easy to learn
- Language with simple rules
- Good for beginners
- Code is readable
- Less development time
- Great support for building web apps
- Dynamic language and no type checking

Limitations in Python

Python is not a good choice for:

- Memory intensive and computation intensive tasks.
- Embedded Systems where processor has limited capacity.
- For graphic intensive 3D game that takes up a lot of CPU.
- Applications that demand concurrency and parallelism
- Developing mobile apps
- Interpreted language and is slow compared to C/C++ or Java

Transition from Python to C/C++

- Will not be so hard
- There are quite a few syntax differences between the two languages
- Only way to learn a new programming language is by writing programs in it

– Dennis Ritchie

Types of Programming Language

- Low level Programming Language
  - **Machine Language** – Machine code- only electronic machines such computers, mobiles etc.. understands this – In 0's and 1's format.
  - **Assembly language** – Directly connected with Computer Architecture.
- High level Programming Language
  - **C**
  - **C++**
  - **JAVA**
  - **Python**

https://www.mentofacturing.com/vincent/implementations.html

C is a language which is directly communicating/working with systems at very low level and interact with computer hardware's.

Simple Problems at a Glance

ABC company Ltd. is interested to computerize the pay calculation of their employee in the form of Basic Pay, Dearness Allowance (DA) and House Rent Allowance (HRA). DA and HRA are calculated as certain % of Basic pay(For example, DA is 80% of Basic Pay, and HRA is 30% of Basic pay). They have the deduction in the salary as PF which is 12% of Basic pay. Propose a computerized solution for the above said problem.

Simple Problems at a Glance

Input : Basic Pay // Getting from the user

Process : Calculate Salary // Calculated by machine

= Basic Pay + (Basic Pay * 0.8) + (Basic Pay * 0.3) - (Basic Pay * 0.12)

----------allowances ---------- --- deductions----

Output : Salary

Simple Problems at a Glance

Little Bob loves chocolate, and he goes to a store with **"Rs. N"** in his pocket. The price of each chocolate is **"Rs. C"**. The store offers a discount: for every **"M"** wrappers he gives to the store, he gets one chocolate for free. This offer is available only once. **How many chocolates does Bob get to eat?**

Simple Problems at a Glance

Input :

- Amount in hand, N
- Price of one chocolate, C
- Number of wrappers for a free chocolate, M

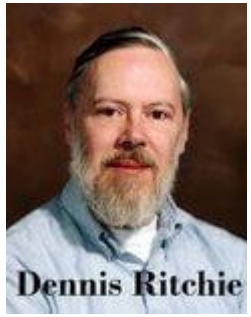
Process :

- Number of Chocolates P = Quotient of N / C
- Free chocolate F = Quotient of P/M

Output :Total number of Chocolates got by Bob

C Programming- Fundamentals

- **What is C?**
  - C is a programming language developed at AT & T's Bell Laboratories of USA in 1972.
  - It was designed and written by Dennis Ritchie.
  - Structured and machine independent language.
  - Unix is one of the most popular network operating system and it was coded entirely in C
  - Reliable and Friendly language
  - What's next?? **C->C++->Java**



History of C Language

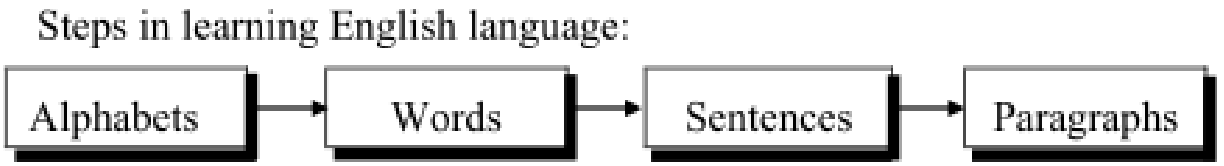
Language	Year	Developed By
Algo	1960	International Group
BCPL	1966	Martin Richard
Traditional C	1969	Ken Thompson
K & R C	1972	Dennis Ritchie
ANSI C	1978	Kernighan & Dennis Ritchie
ANSI/ISO C	1989	ANSI Committee
C90	1990	ISO Committee
C99	1999	ISO Committee
C11	2011	Standardization Committee
C18	2018	Standardization Committee

C Programming- Fundamentals

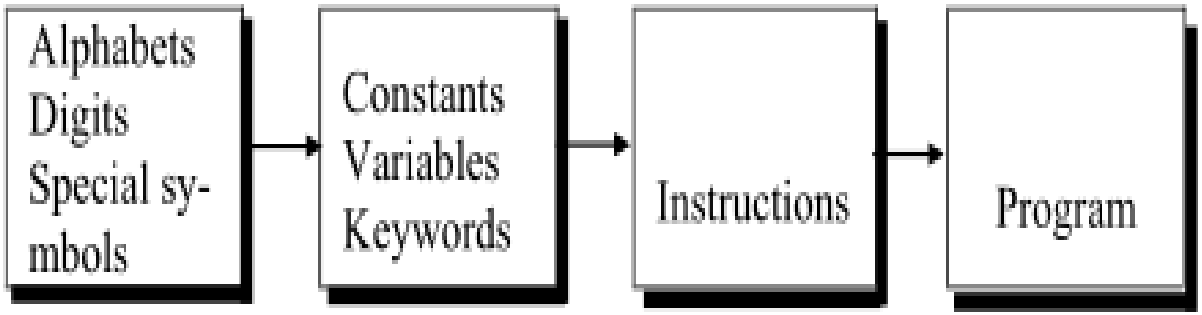
- **Features of C**
  - General-purpose Structured language
  - Free format and case-sensitive language
  - System Programming language
  - Fast running and efficient language
  - Machine independent and hence it is portable
  - Commands can be inserted anywhere in the program
  - Memory address are directly accessed by using pointers
  - Supports modularity
  - Permits recursion
  - More built-in functions

Getting Started with C

- Communicating with a computer involves speaking the language the computer understands.
- What is the learning method of English??



- Steps in learning C



World in "C"

- OS – Windows / UNIX / LINUX
- Device Drivers Programs
- Mobile Devices
- Microprocessor in washing machine, digital camera, microwave oven. Embedded C
- 3D Computer Games - still written in C

C is always best when it comes to performance(*in terms of speed of execution*)

C is the building block of all the programming languages

What's Next?? ……Implementation

- **IDE**
- **Single platform** where you get numerous features required such as text editor, syntax highlighter, customizable interfaces, compiler, code auto-save,debugger, build automation, and deployment.
- **Why IDE?**
  - Writing ***programs easy, efficient, and effective*.**
  - It saves lot of time.
  - Freedom to **choose** the **programming language of your interest.**
- Eclipse, Visual stdio, Netbeans, Clion, **Code Blocks,** Codelite, Qtcreater etc….

How To Install and Getting Started

- CodeBlocks is an open-source, cross-platform (Windows, Linux, MacOS), and free C/C++ **IDE**.
- ***Website :*** **www.codeblocks.org/downloads**

Step 1: Download

- Go to http://www.codeblocks.org/downloads. Click "Download the binary release".
- Select your operating platform (e.g., Windows XP/Vista/7/8.x/10 **or** Linux 32 and 64-bit **or** Mac OS X).
- Download the installer with GCC Compiler(codeblocks-20.03mingw-setup.exe) (which includes MinGW's GNU GCC compiler and GNU GDB debugger).

Step 1: Download

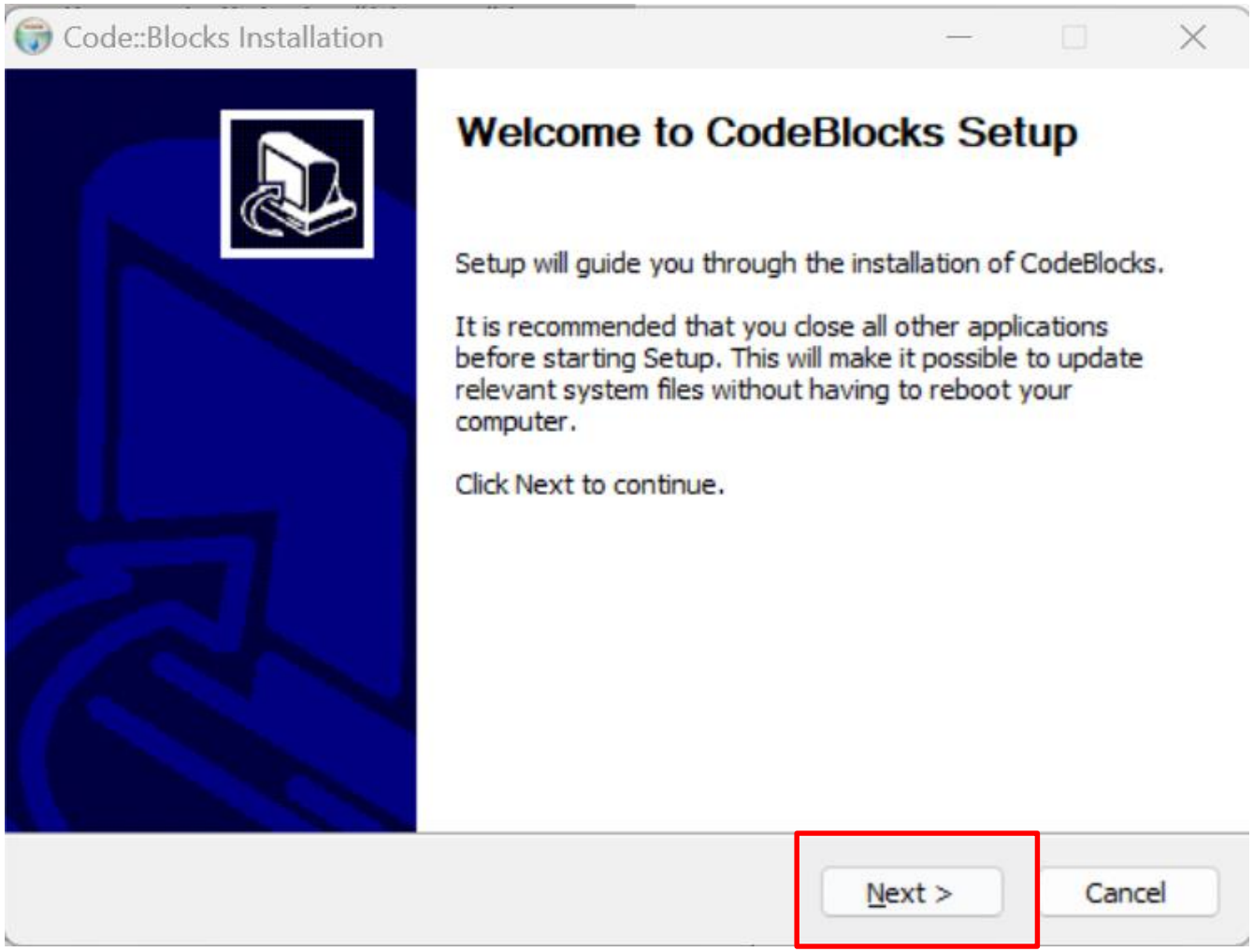
Microsoft Windows

File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

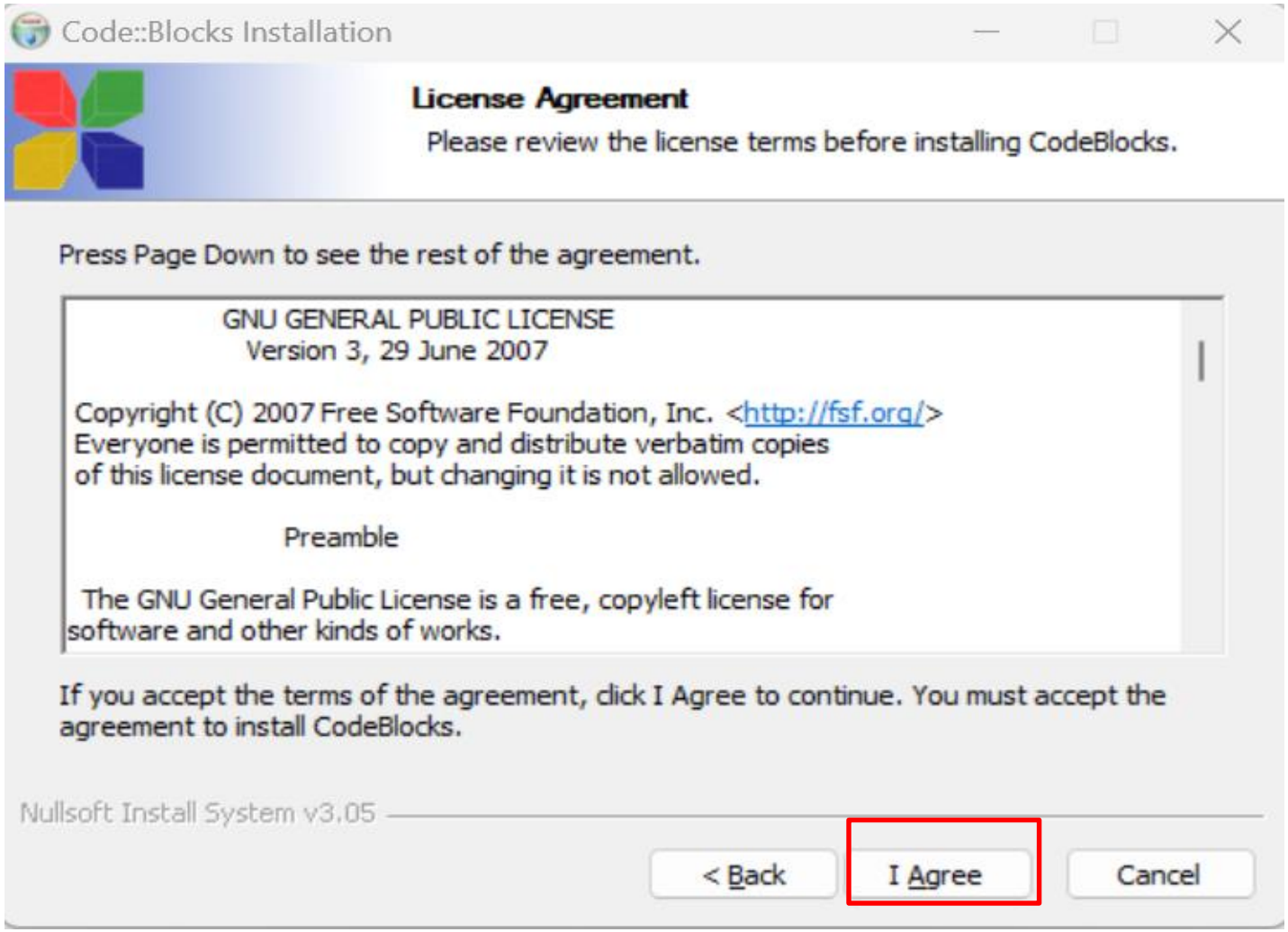
NOTE: The codeblocks-20.03-setup.exe file includes Code::Blocks with all plugins. The codeblocks-20.03-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their machine(s).

NOTE: The codeblocks-20.03mingw-setup.exe file includes additionally the GCC/G++/GFortran compiler and GDB debugger from MinGW-W64 project (version 8.1.0, 32/64 bit, SEH).

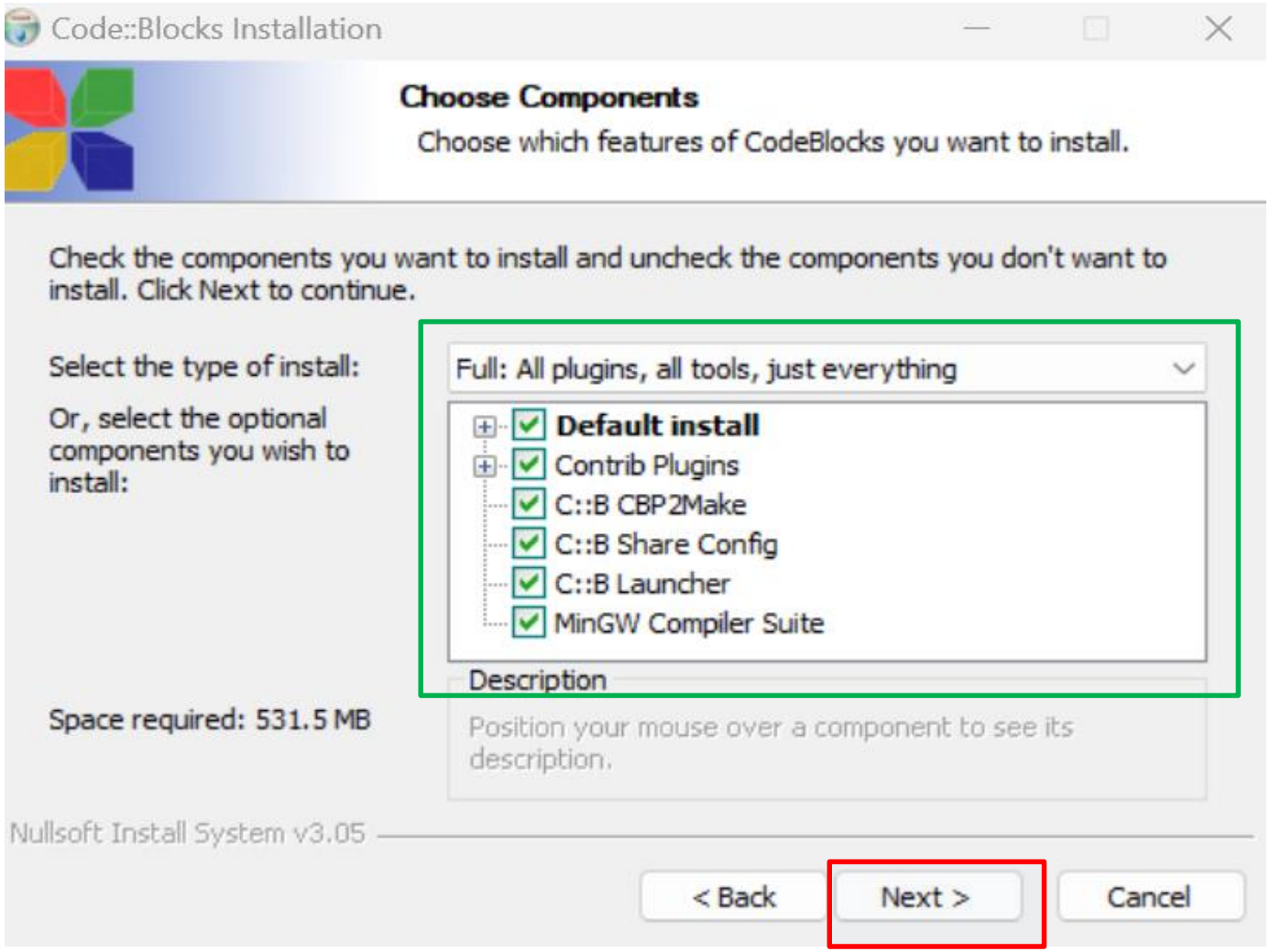
Step 2: Double-click to run the downloaded installer and click the “Next >” button.



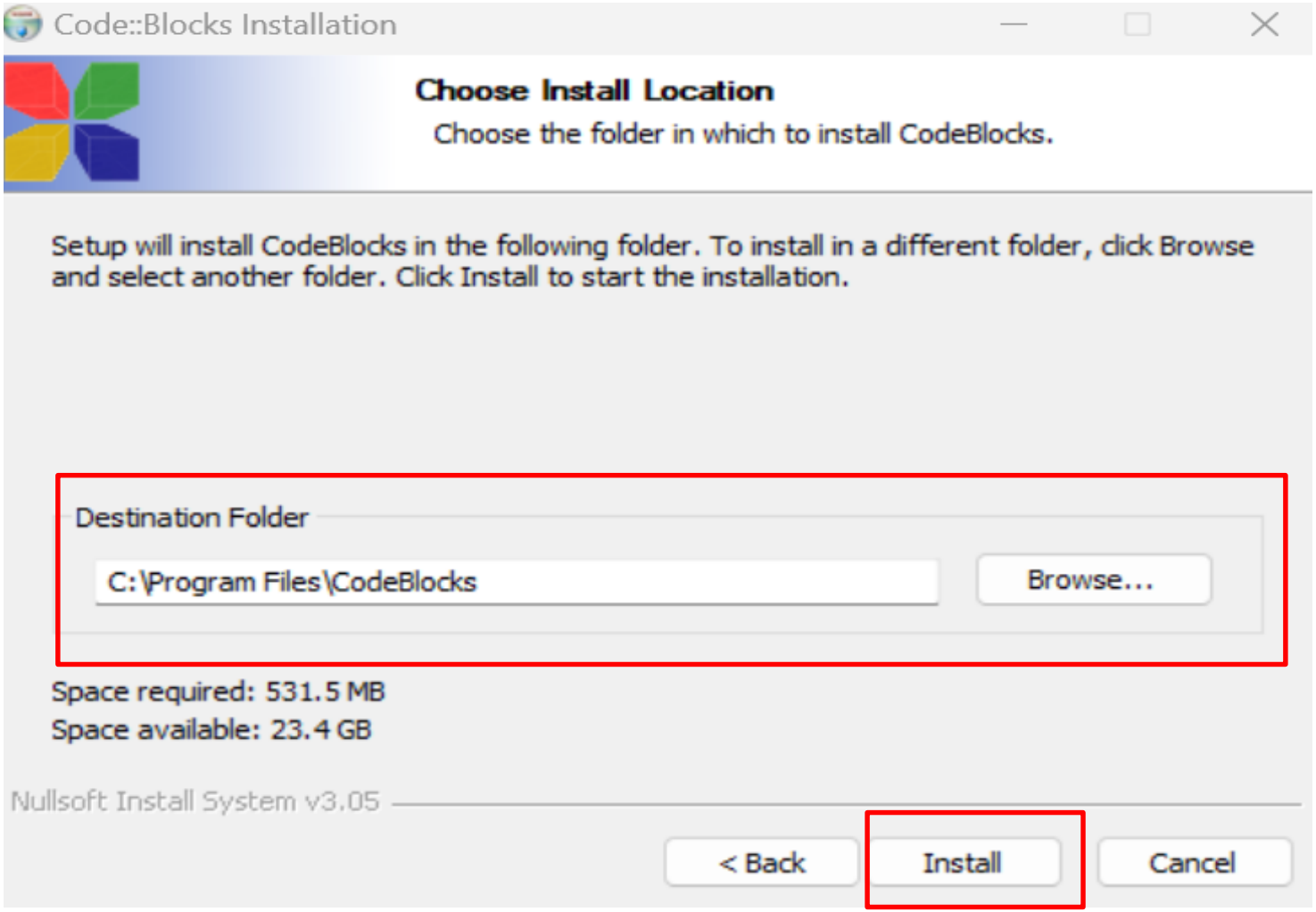
Step 3: Now click “I Agree” to accept the license agreement.



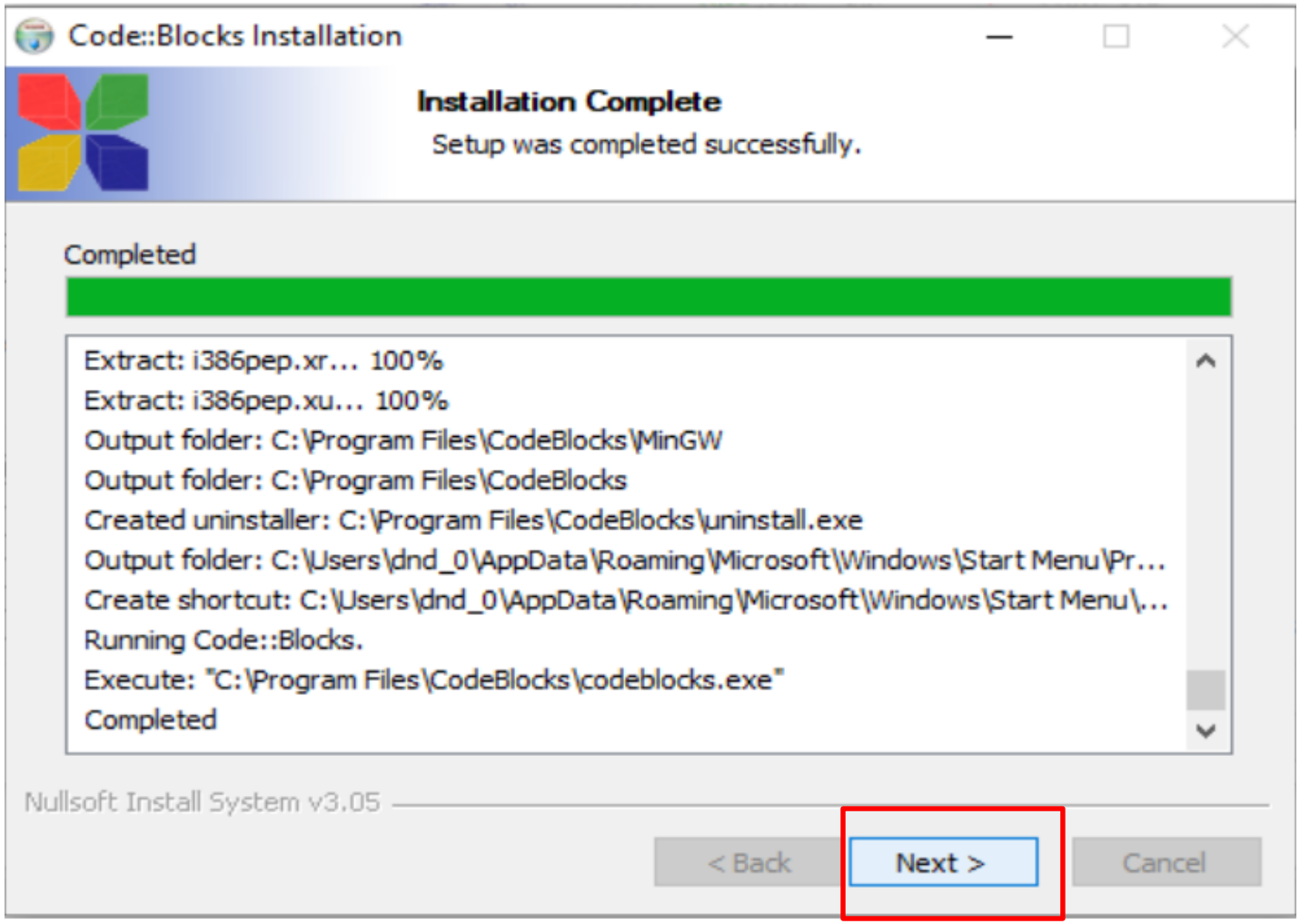
Step 4: The next step, don't do anything, just click the "Next >" button to install packages.



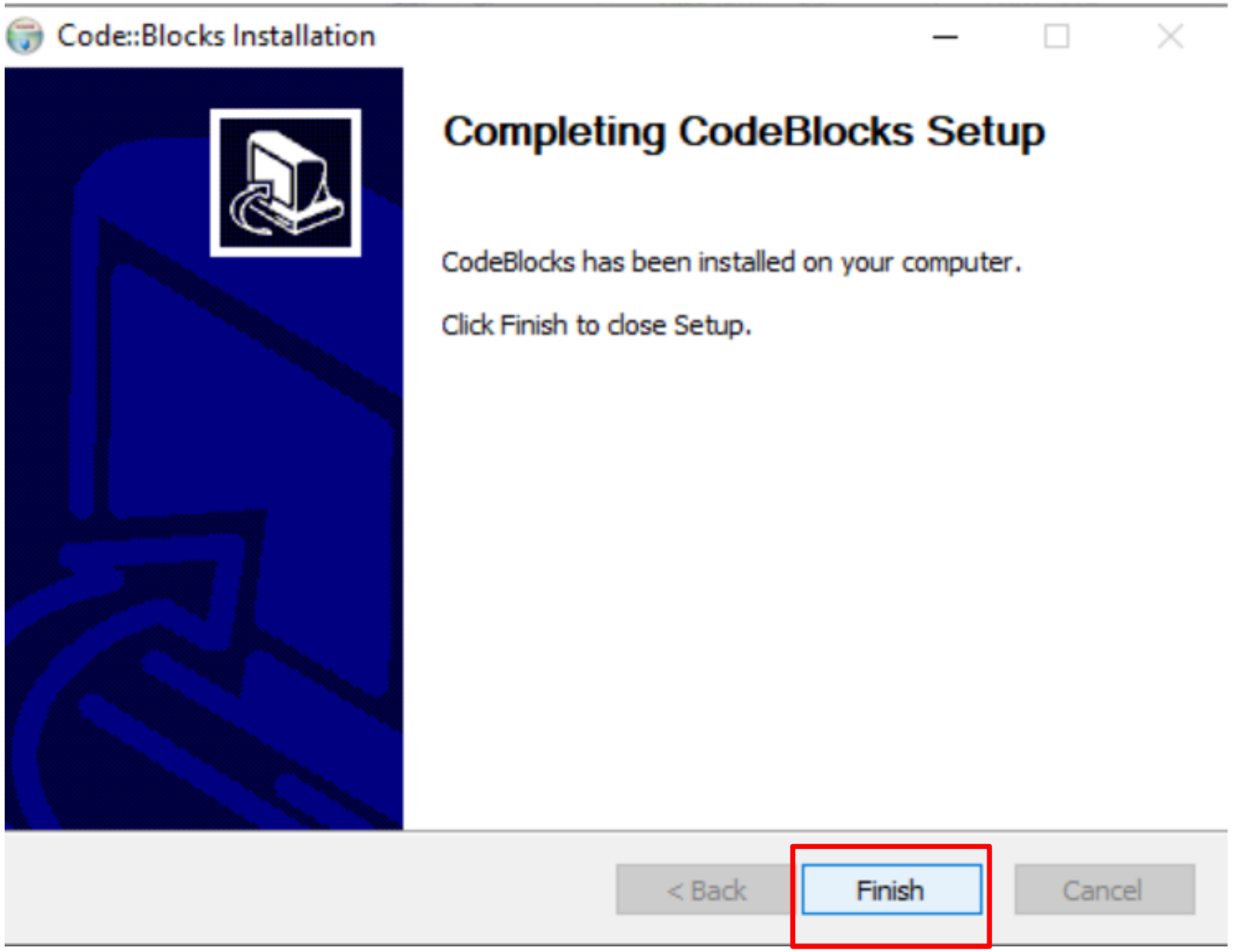
Step 5: On the next window, then change the installation directory if you want (default directory recommended). Click the "Install" button.



Step 6: After the installation is completed click the “Next >” button..

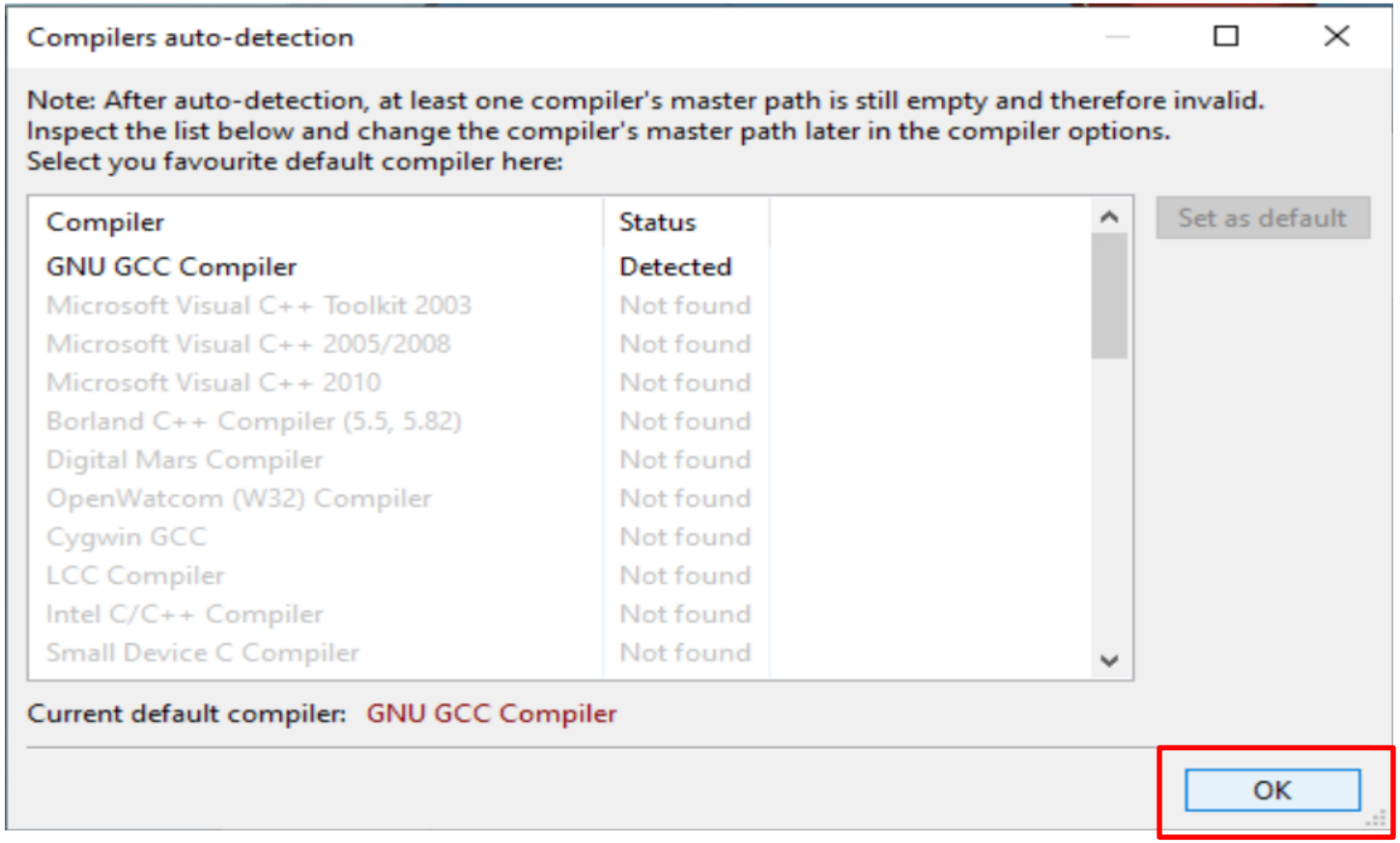


Step 7: To complete the installation process click the “Finish” button.



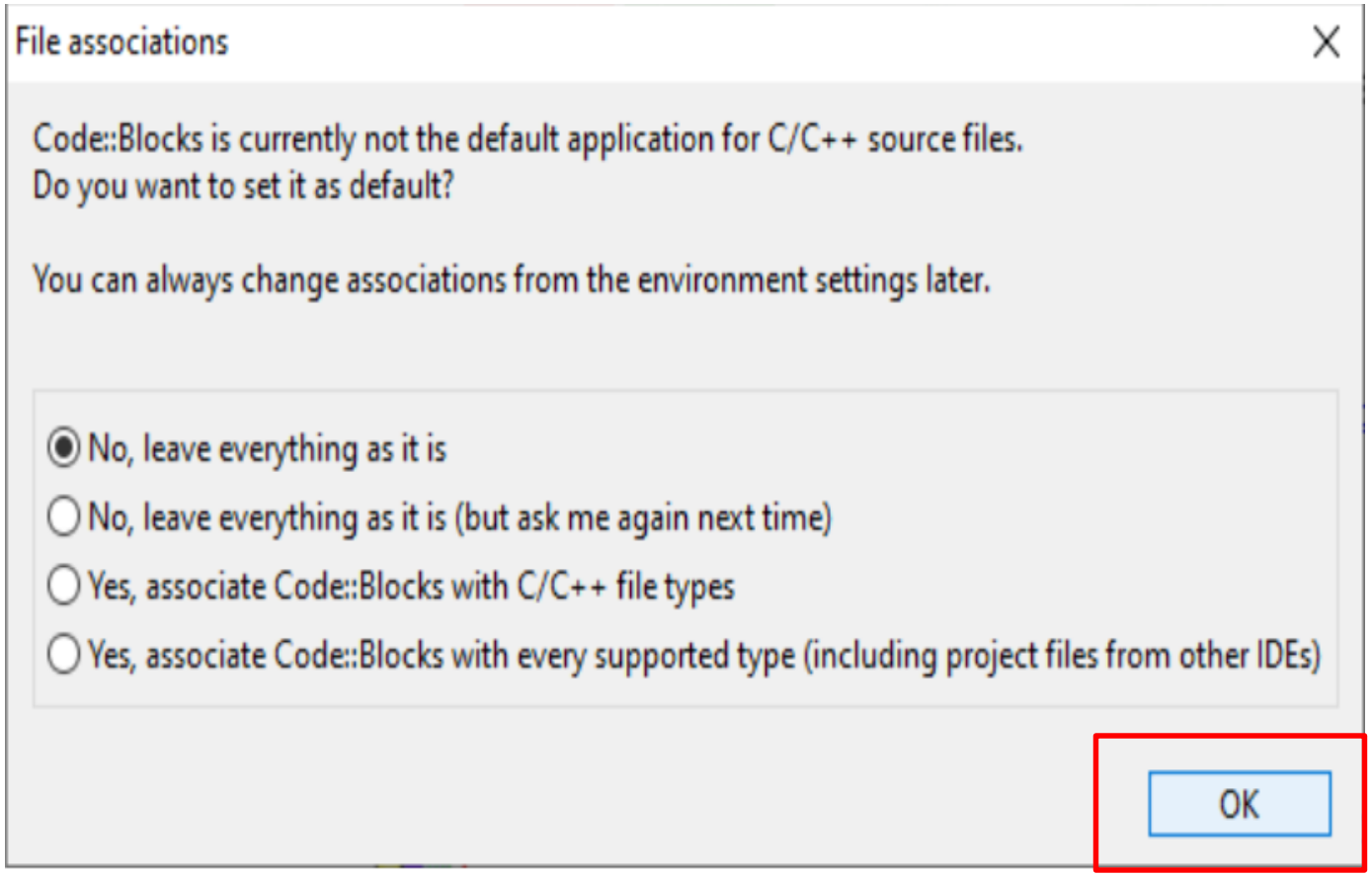
RUNNING AND CONFIGURATION

- Go to Start Menu > All Apps > CodeBlocks > CodeBlocks.
- In the window "Compiler auto-detection", just click the "Ok" button.



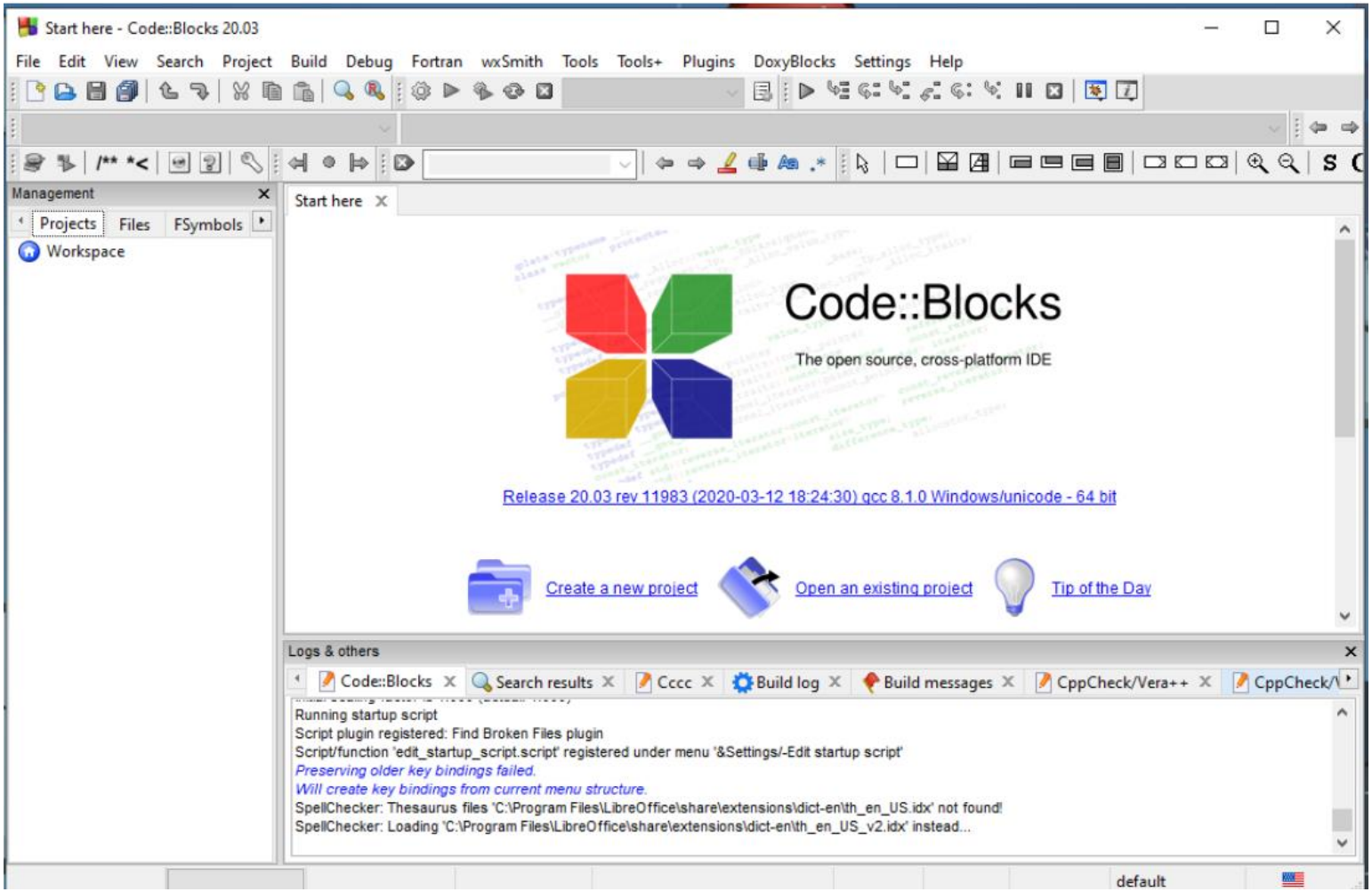
RUNNING AND CONFIGURATION

- Make a choice for the file association and click on the "OK" button



RUNNING AND CONFIGURATION

- Final coding Window

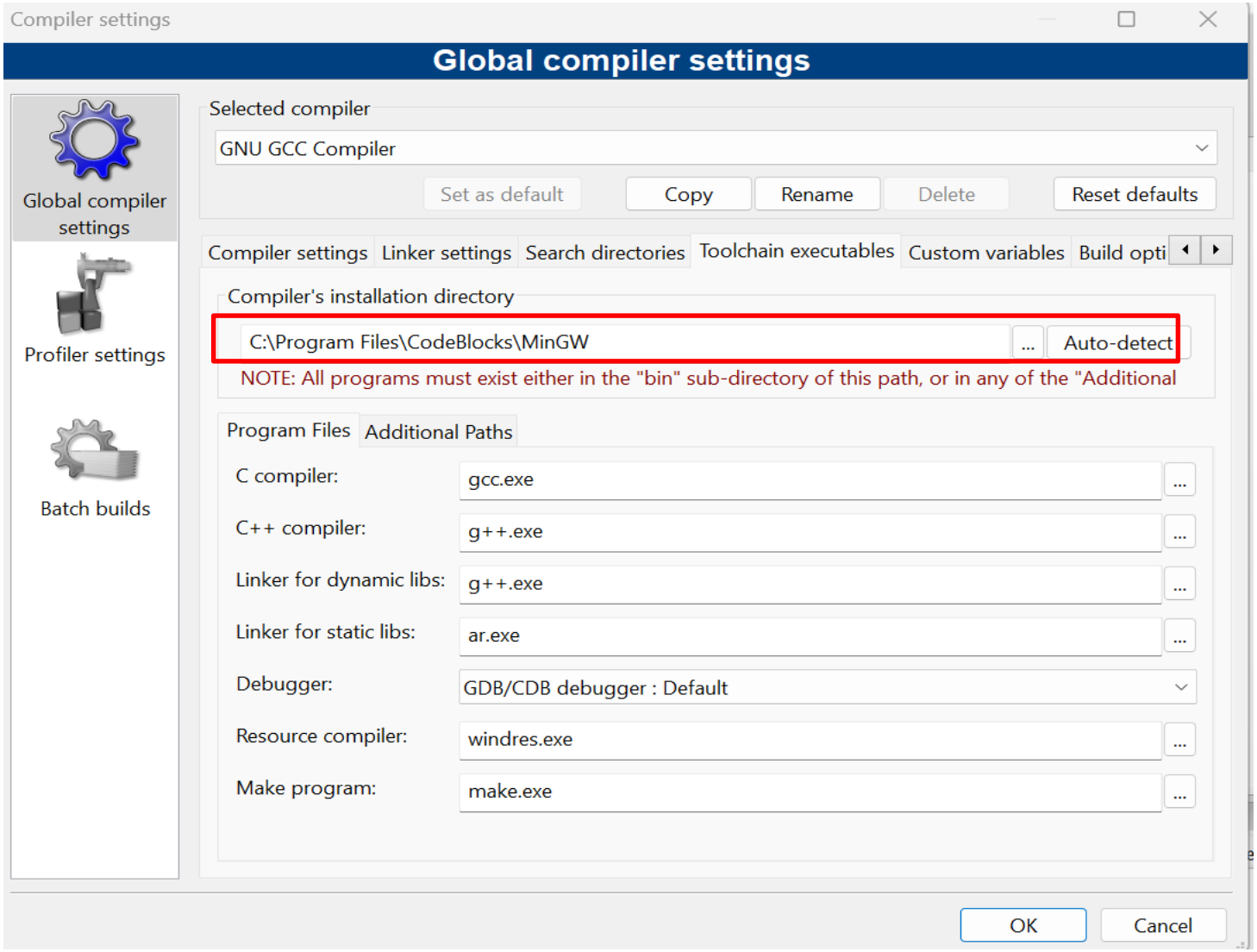


RUNNING AND CONFIGURATION

- **Verify the Compiler's and Debugger's Path:**
- Goto "Settings" menu ⇒ "Compiler..." ⇒ In "Selected Compiler", choose "GNU GCC Compiler"
- Select tab "Toolchain Executables" ⇒ Check the "Compiler's Installation Directory". It shall be set to the "MinGW" sub-directory of the CodeBlocks installation directory, for example, suppose that CodeBlocks is installed in **"c:\Program Files\codeblocks", set it to "c:\Program Files\codeblocks\MinGW".**

RUNNING AND CONFIGURATION

- **Verify the Compiler's and Debugger's Path:**



RUNNING AND CONFIGURATION

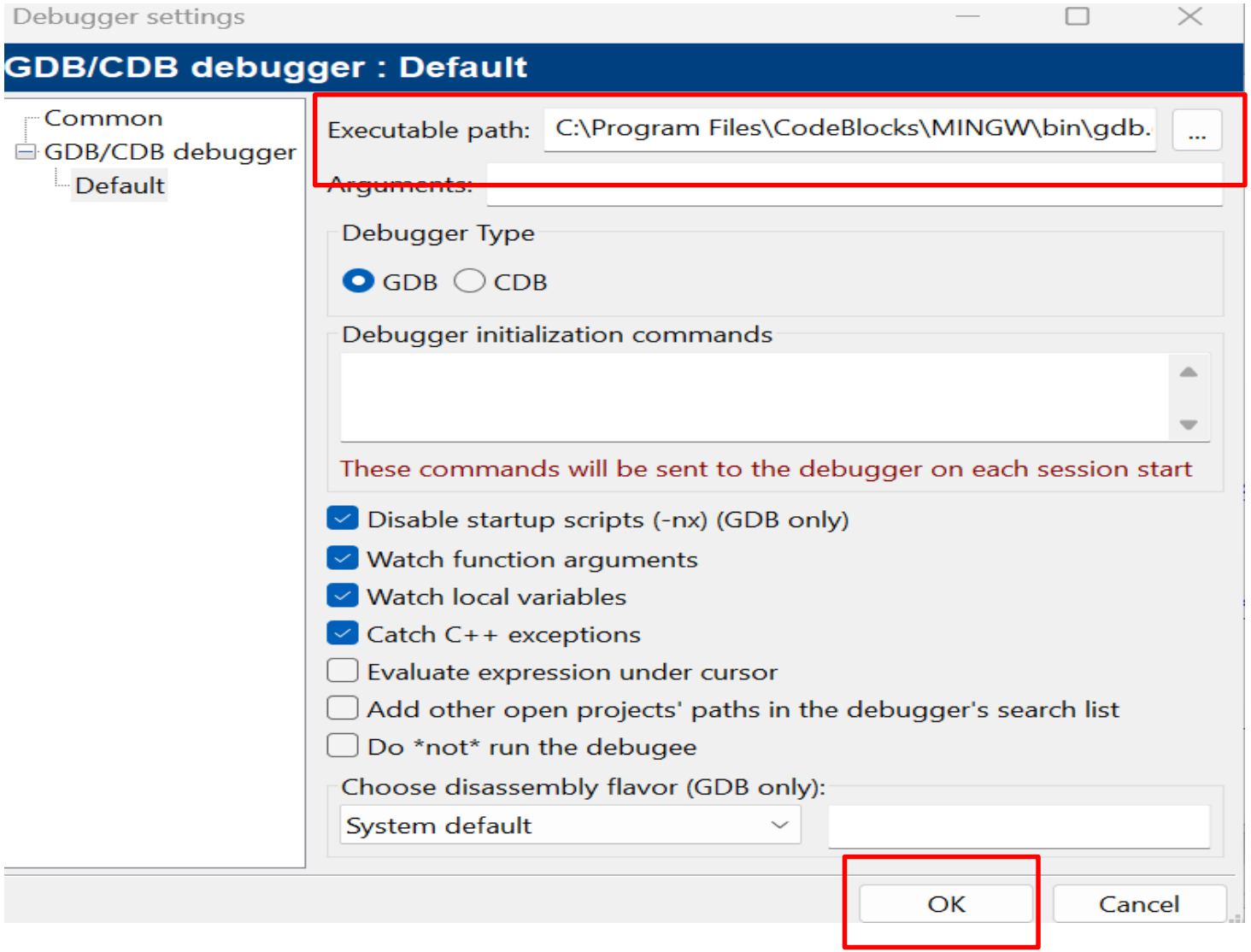
- **Checking of Debugger's Path:**

Goto "Settings" menu ⇒ "Debugger..." ⇒ Expand "GDB/CDB debugger" ⇒ Select "Default" ⇒ In "Executable path", provide the full-path name of "gdb.exe",

For example, "c:\ProgramFiles\codeblocks\MinGW\bin\gdb.exe".

RUNNING AND CONFIGURATION

- **Checking of Debugger's Path:**



Writing C/C++ Program in Code blocks

- **For Simple Programming Exercise**

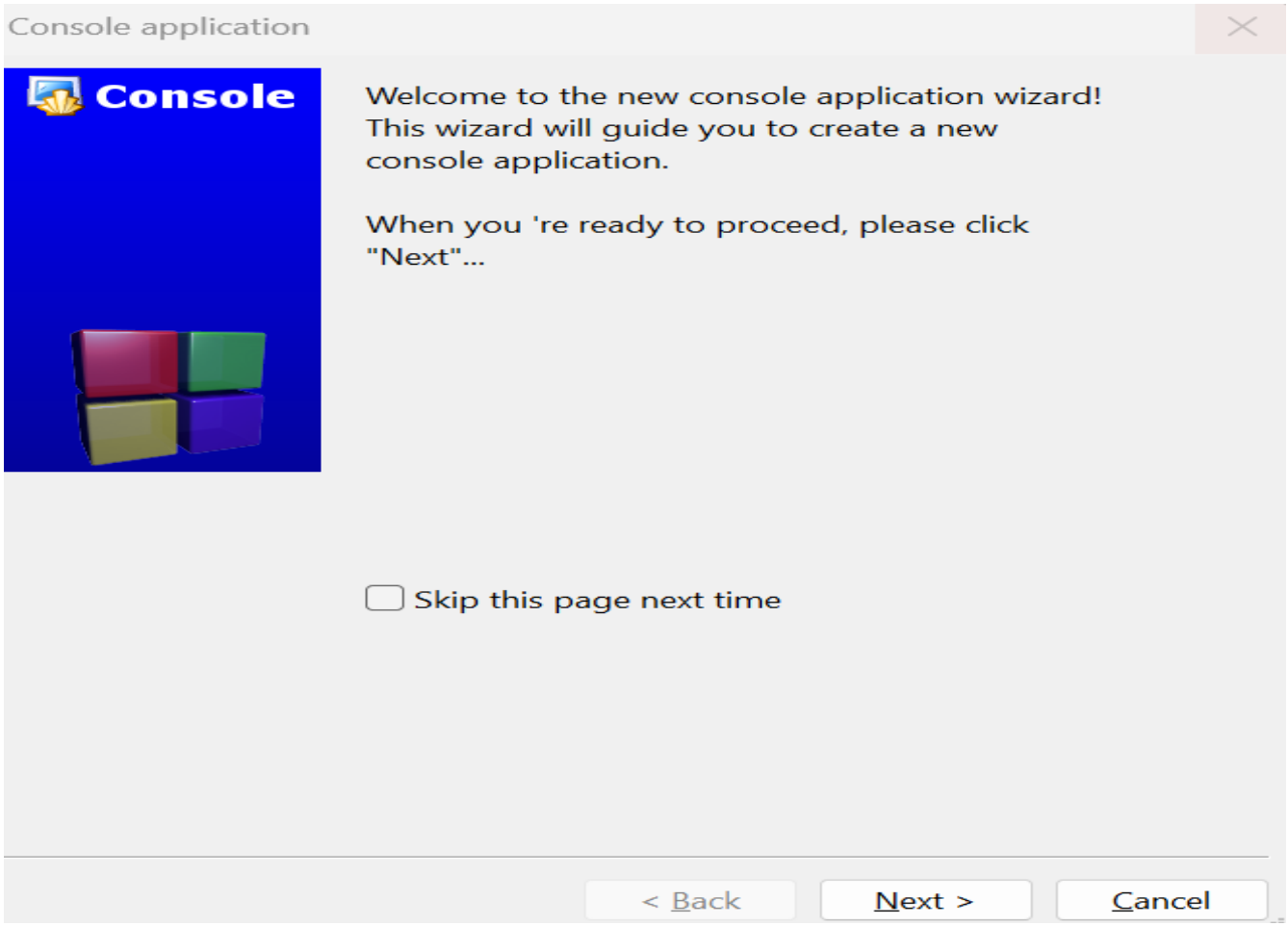
- File ⇒ New ⇒ **Empty File**.
- Enter the Code
- Save the file as "test1.c" in your own directory (e.g., "*D:\IYAPPANDATA\VIT\NOV2022 - APRIL 2023\C&C++*").
- Build (Compile and Link): Select "Build" menu ⇒ Build (Ctrl-F9).
- Run: Select "Build" menu ⇒ Run (Ctrl-F10).
- Build and Run=> F9

Writing C/C++ Program in Code blocks

- **Writing Programs(Under Project)**
  - A project contains related files such as source codes, header files, and relevant resources. Also, under CodeBlocks, you can only debug your program under a project - single-file program (in previous section) debugging is not supported.
  - File ⇒ New ⇒ Project... ⇒ Console Application ⇒ Go. "Console Application" wizard appears => Next

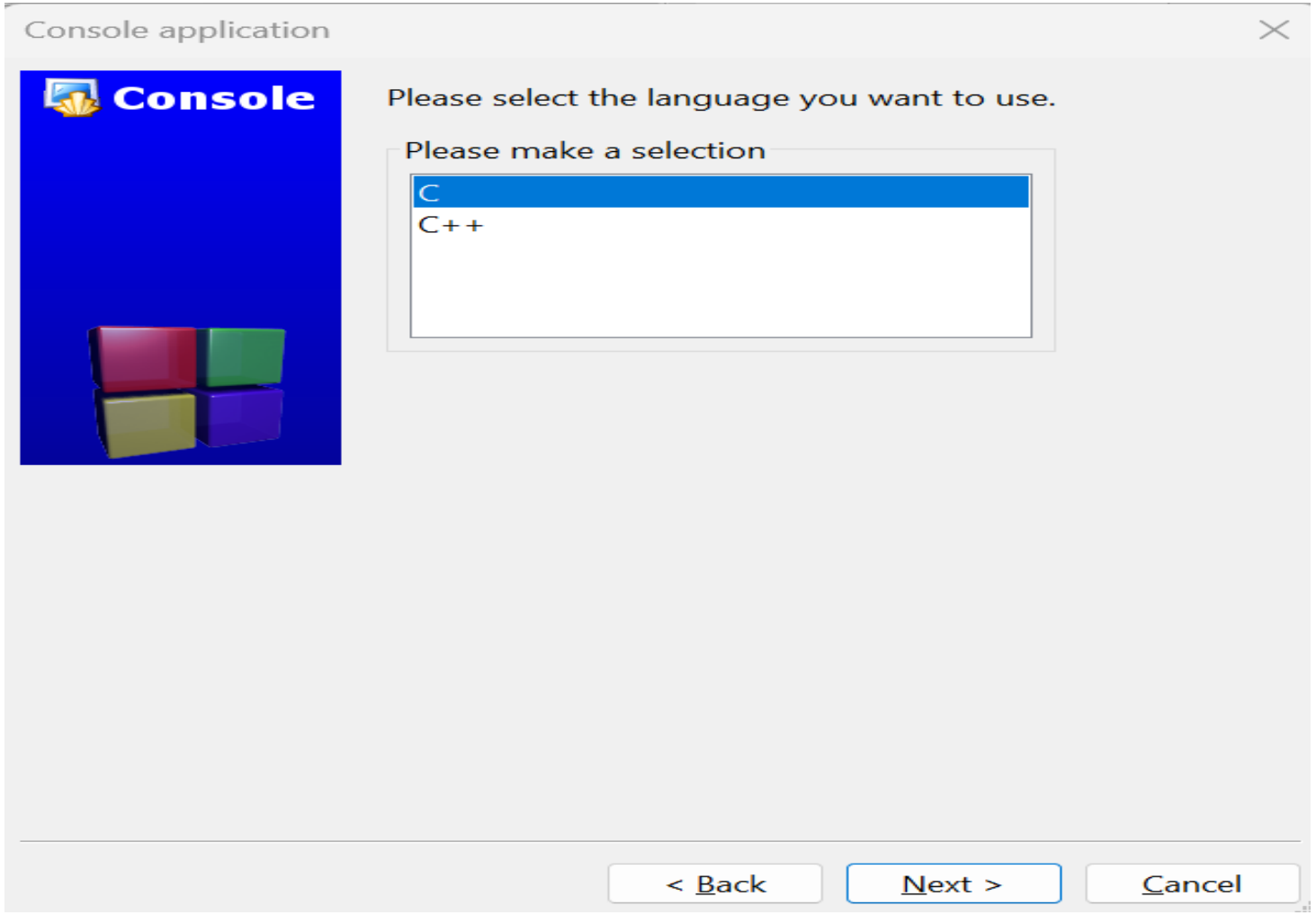
Writing C/C++ Program in Code blocks

- **Writing Programs(Under Project)**



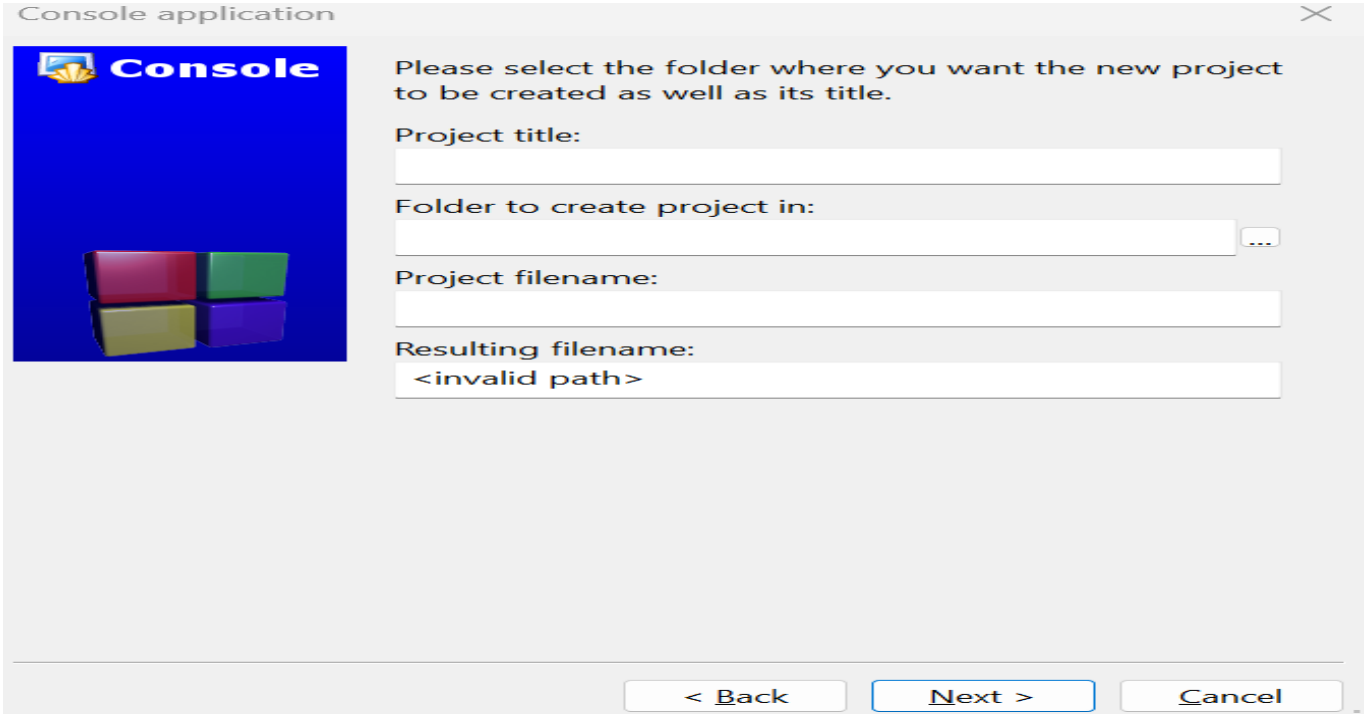
Writing C/C++ Program in Code blocks

- **Writing Programs(Under Project)**



Writing C/C++ Program in Code blocks

- **Writing Programs(Under Project)**



In "Project Title", enter “ProjectName”. In "Folder to create project in", set to your working directory, e.g., "*D:\IYAPPANDATA\VIT\NOV2022 - APRIL 2023\C&C++*"". Accept the default for the rest ⇒ Next.

Writing C/C++ Program in Code blocks

- **Writing Programs(Under Project)**

- Under the "Management" pane ⇒ Choose "Projects" tab ⇒ Expand the project node "ProjectName" ⇒ Expand "Source" node ⇒ Double-click "main.c", which is a template program to say "Hello, world!".

- To build the program, select "Build" menu ⇒ Build.
- To run the program, select "Build" menu ⇒ Run.

Writing C/C++ Program in Code blocks

Create more source file or header file under the project

- File ⇒ New File... ⇒ Select C/C++ source or C/C++ header =>C ⇒ Next.
- In "Filename with full path" ⇒ Click the "Navigate" (...) button to navigate to the project directory and enter the new file name. Check both the "Debug" and "Release" boxes (or "All") ⇒ Finish

Sample Problems to Try

- Find the average runs scored by a batsman in 4 matches.
- Area of a circle
- Compute Simple interest, compound interest.
- Convert Fahrenheit to centigrade and vice-versa.

.

Thank you