

C++ PROGRAMMING



for Second Stage

Prepared by

Asst. Lecturer Wala'a N. Jasim

2020/2021

Outline

- **Program Parts**
- **Variable Types**
- **Error Variables**
- **CodeBlocks**
- **Open CodeBlocks**
- **Steps to Create a New Project in CodeBlocks**
- **Steps to Run The Project Properly**
- **Cout**
- **Cin**
- **Examples**

C++ programming

PROGRAM PARTS

C ++ program consist of three main parts:

- **Header Files**
- **Variables and declarations**
- **Program body**

C++ programming

1) Header Files

 `#include <iostream.h>`

 `#include <conio.h>`

 `main ()`

C++ programming

2) Variables and declarations:

Variable:

A location in the computer's memory the different values are stored in it.

- Int  (1,6. -7
- Float  (16.8 , 7.5 ,.....)
- Char  (a,b,c,..+,=,-,* ,/,...)

C++ programming

TABLE OF VARIABLE TYPES

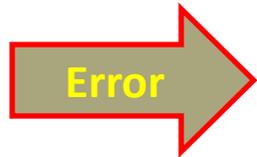
Type	Mean
Bool	Boolean / to store logical values (True or False)
Char	Character/ to store letters ASCII encoding
Int	integer
Float	floating point

C++ programming

Error Variables



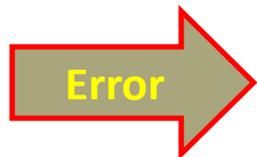
int 1a;



char a ge;



int %;



float do;

C++ programming

3) Program body

In this part processing required is written.

```
#include <iostream.h>
#include <conio.h>
Void main()
}
int a,b;
Cout<<"enter the first number please:";
Cin>>a;
Cout<<"\n enter the second number please:";
Cin>>b;
Cout<<"sum ="<<a+b;
getch();
}
```

Header

Variables and declarations

Program Body

C++ programming

To create a successful program, we must do the following:

- Knowing what to do in the program.
- The inputs used and the type of processing required.
- Knowing the number and type of variables that we will deal with in order to be able to reserve locations for them in the definitions and declarations section.
- Attention to common errors.

C++ programming

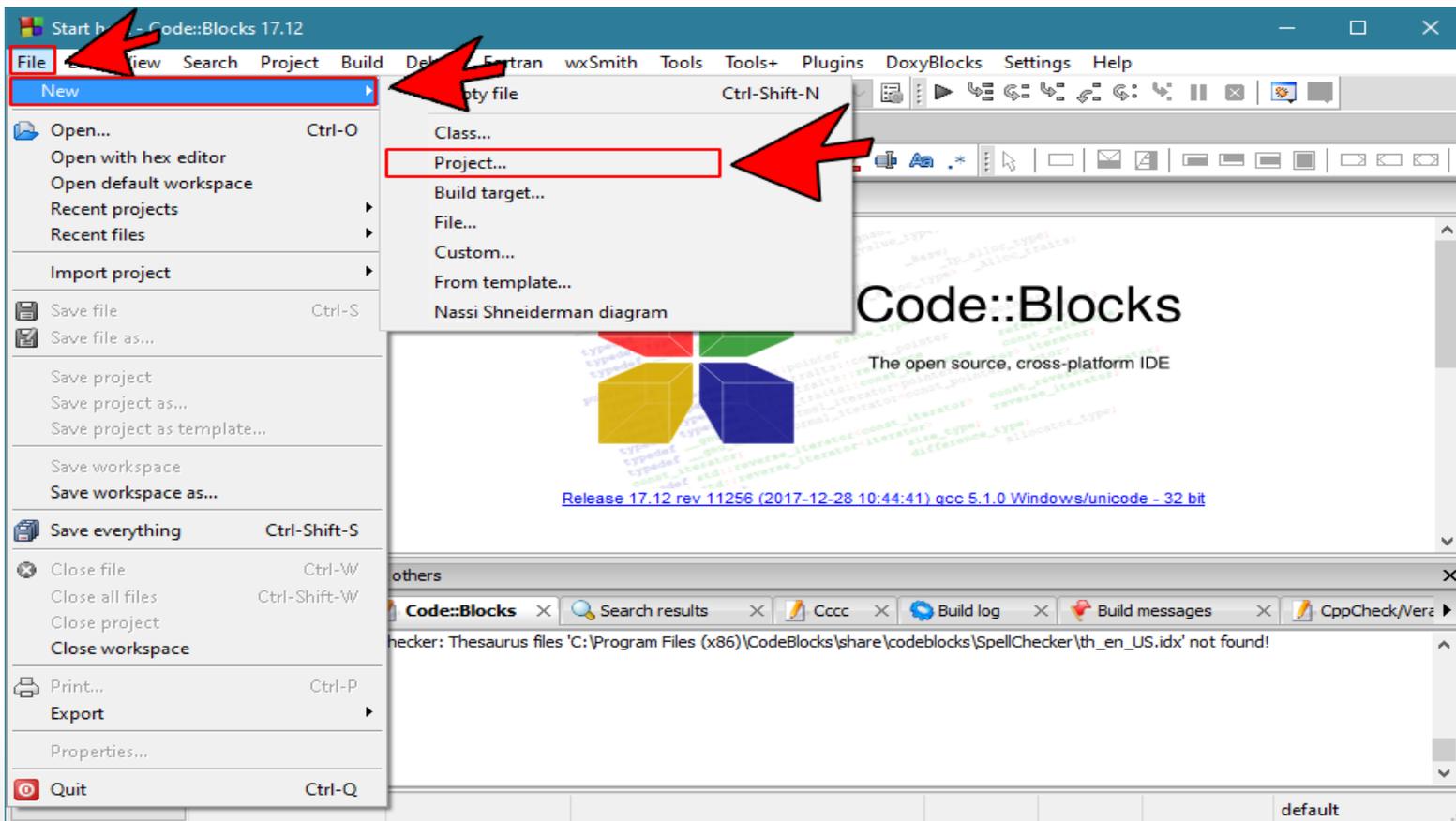


Code::Blocks is a free and open source code for writing code in C and C ++ language. The program is written in C ++ language. The program can be used to write codes in several other languages such as Fortran. CodeBlocks works on Windows, Mac and Linux operating systems; The latest version of the program is 20.03.

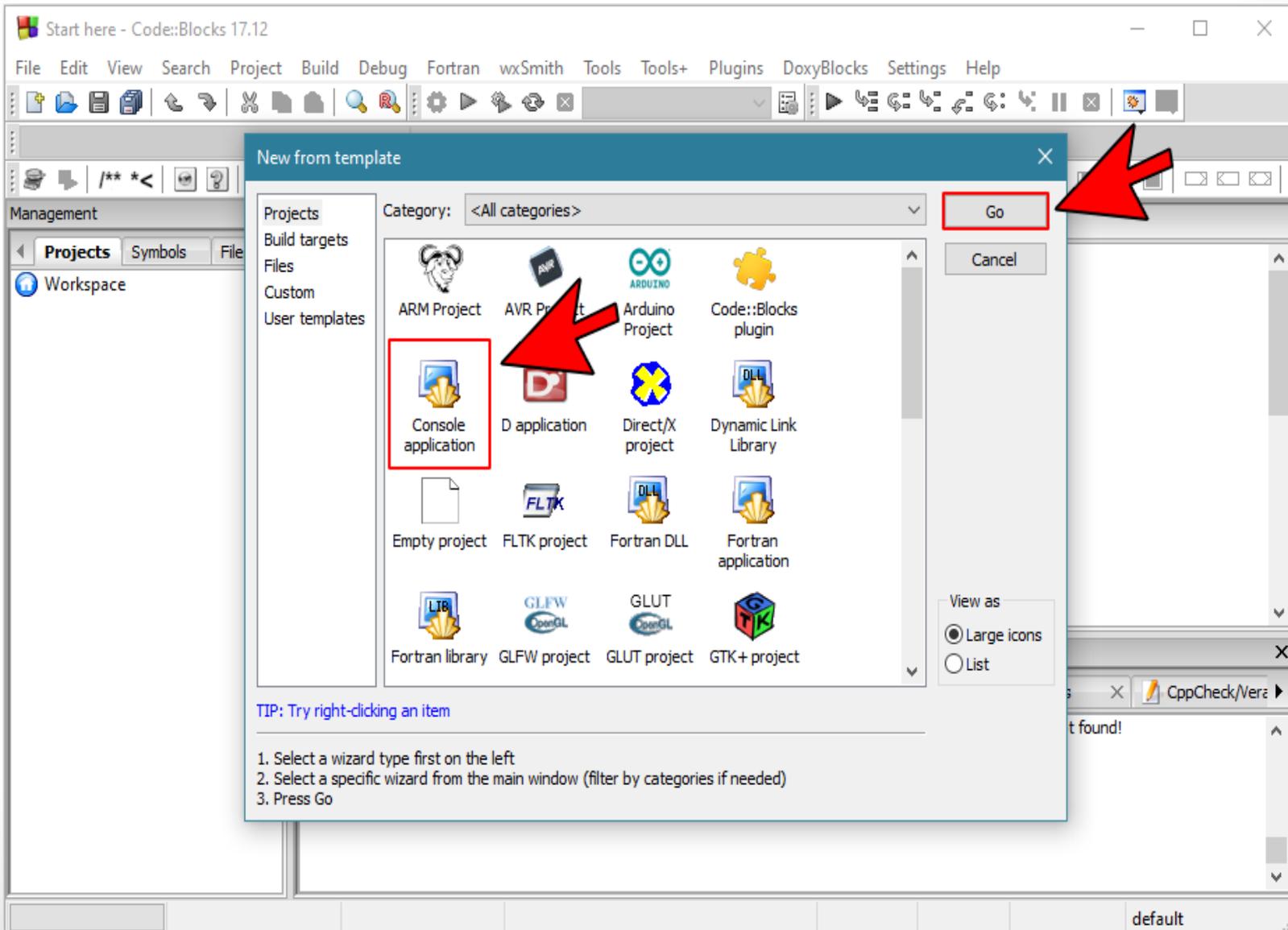
Steps to create a new project in CodeBlocks

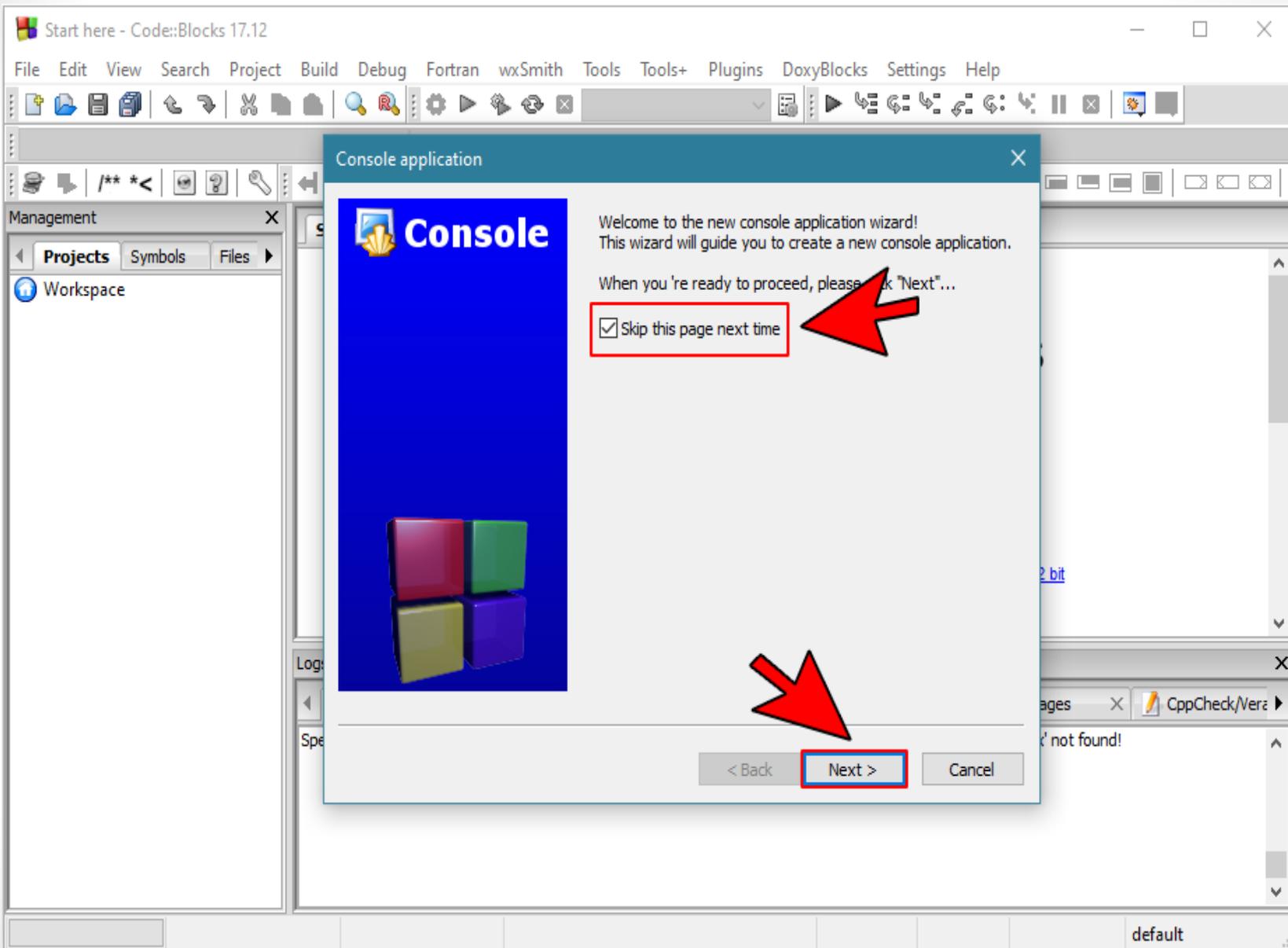
1- Open CodeBlock

2- Choose File → New → Project

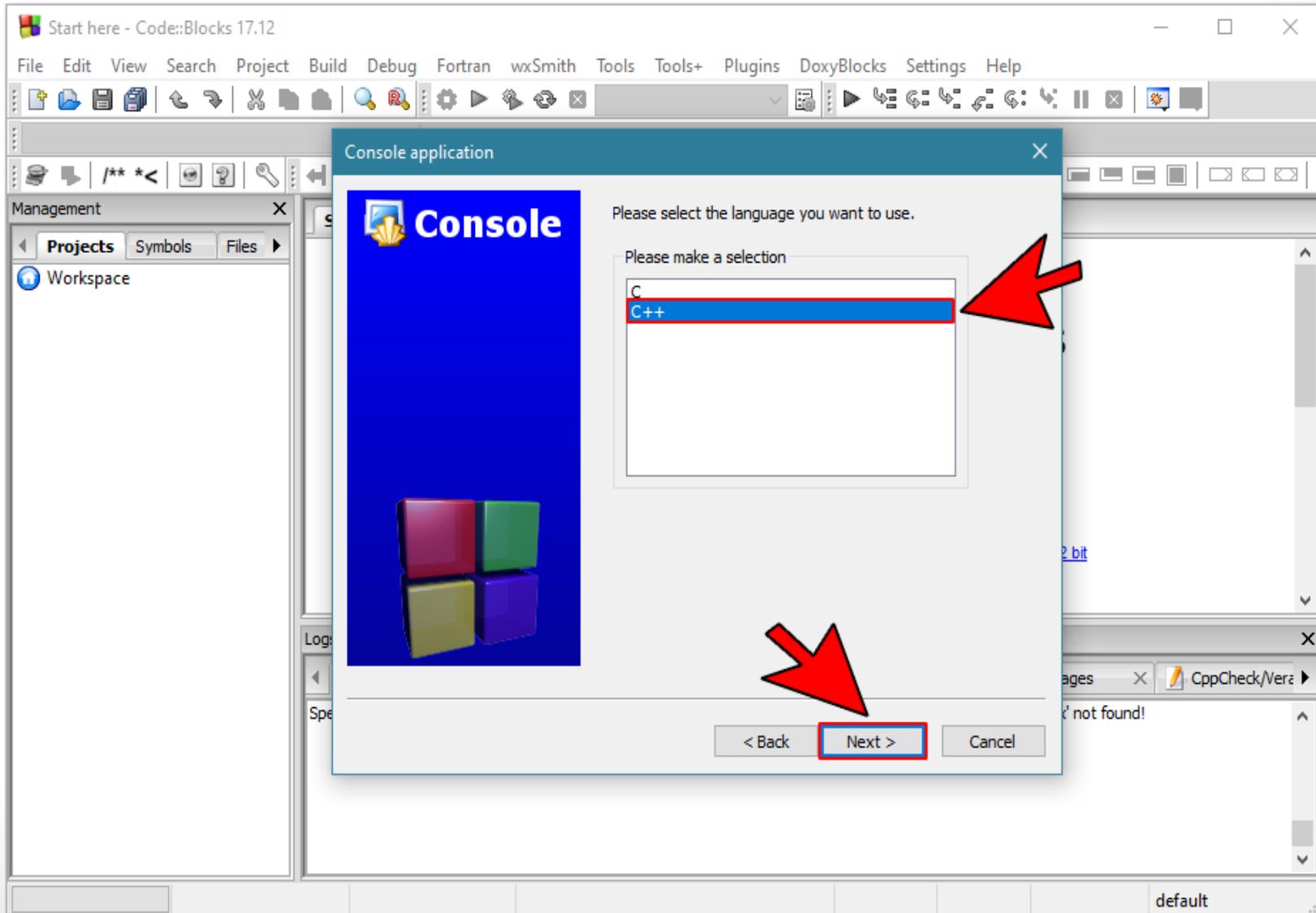


Click Console Application then choose Go

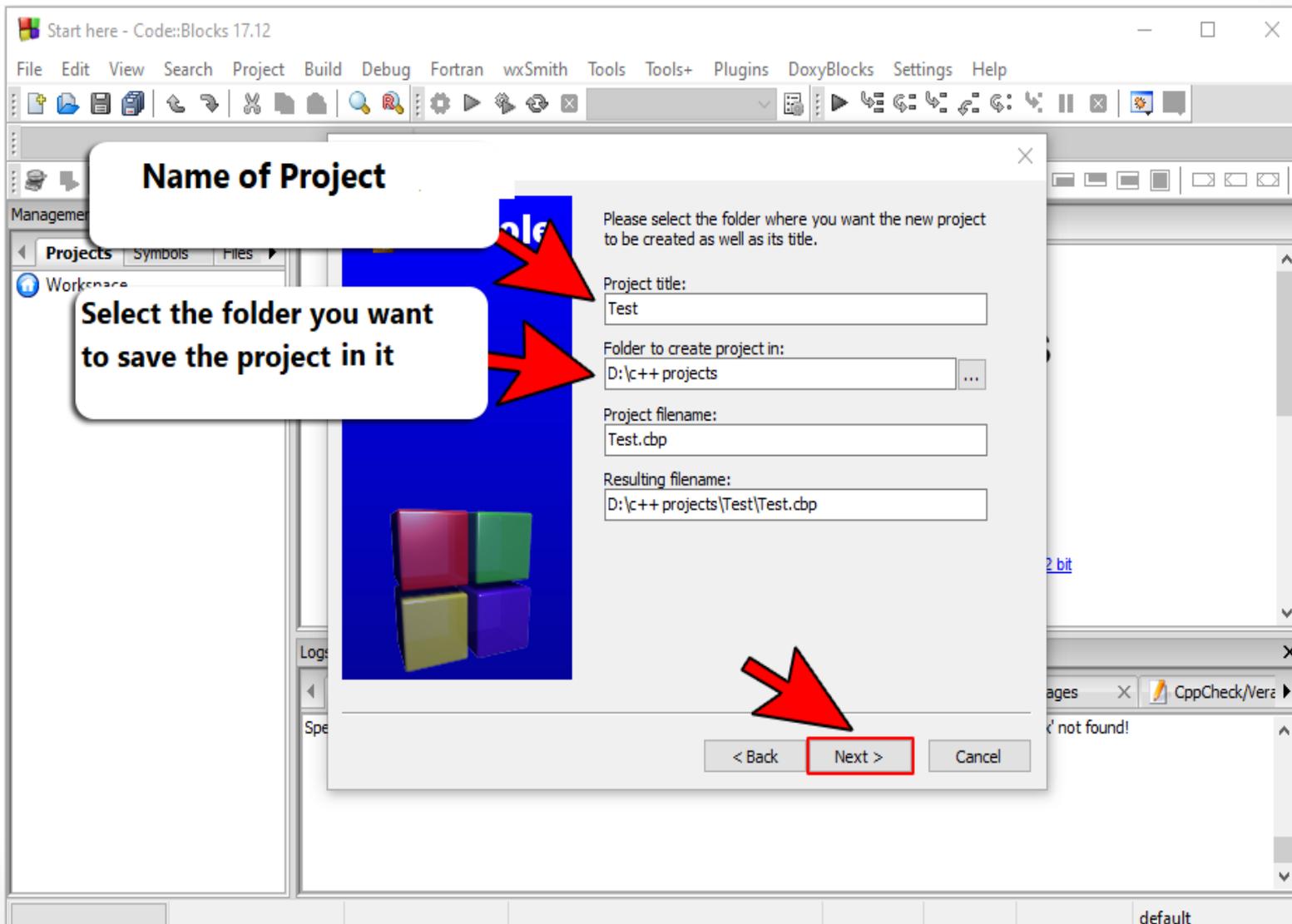




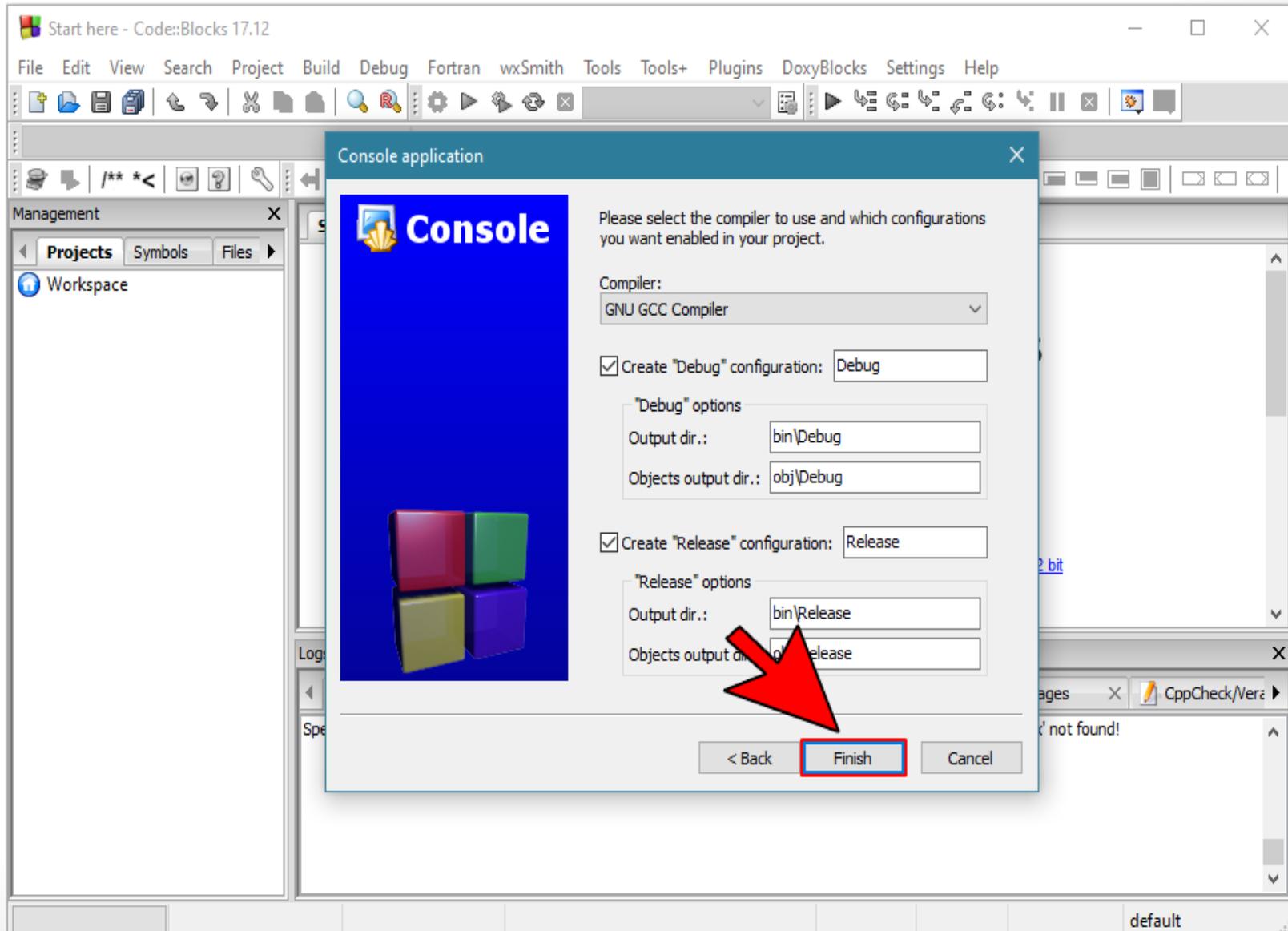
Click C++ then Next



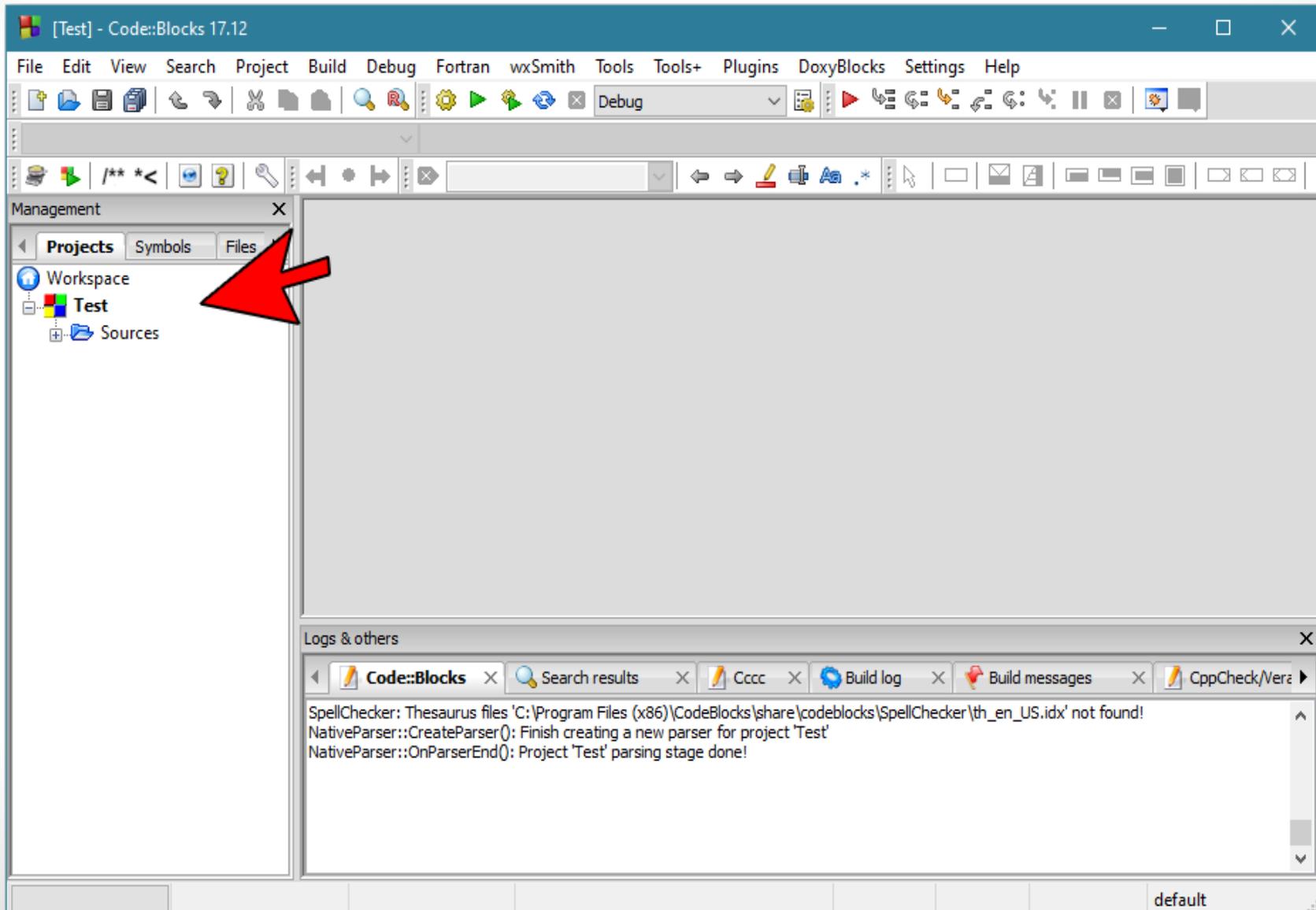
Specify the name you want to put the project and the path of the folder in which you want it to be saved on your computer, then click Next as follows.



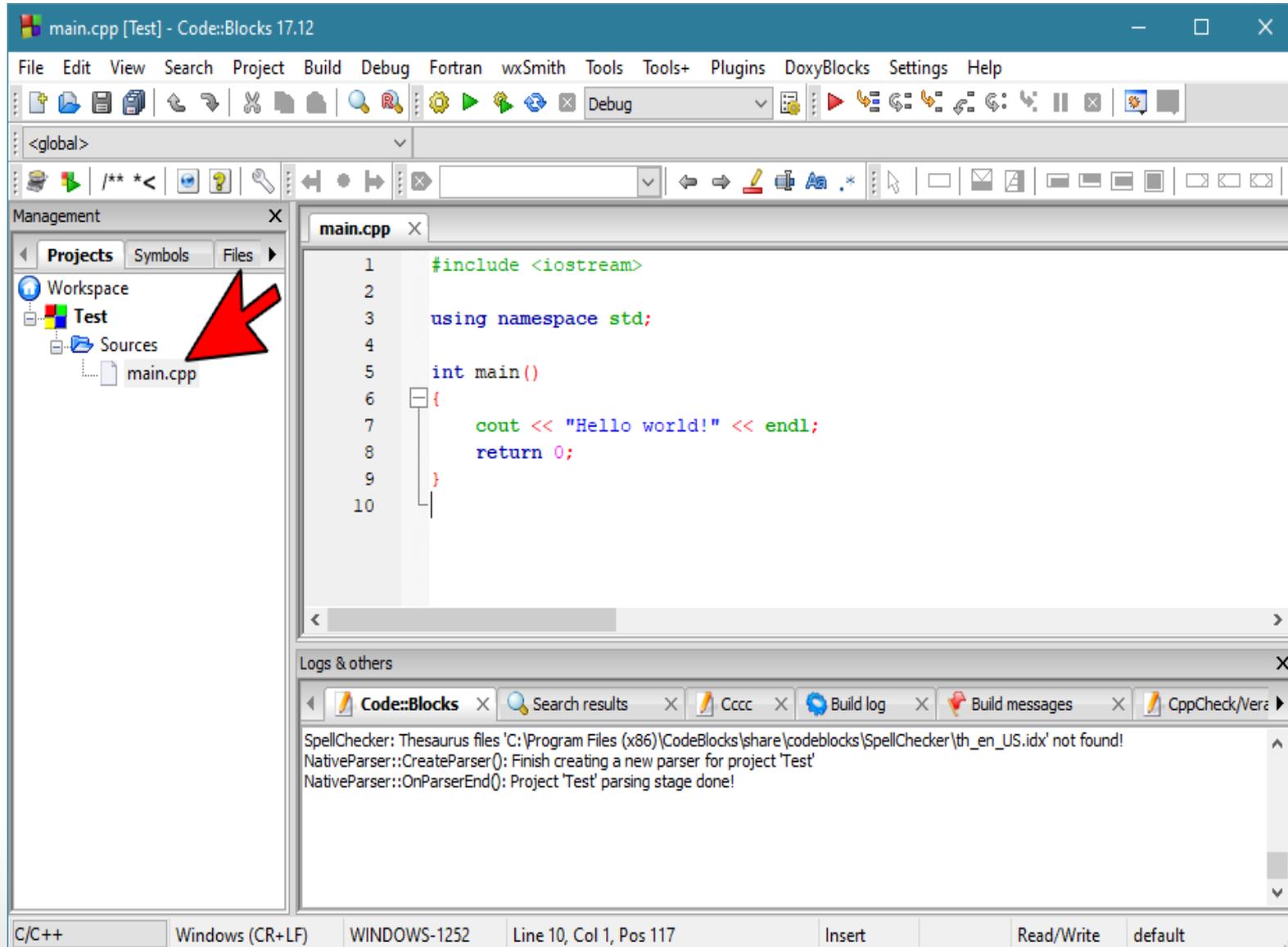
Don't edit anything in this windows then click finsh



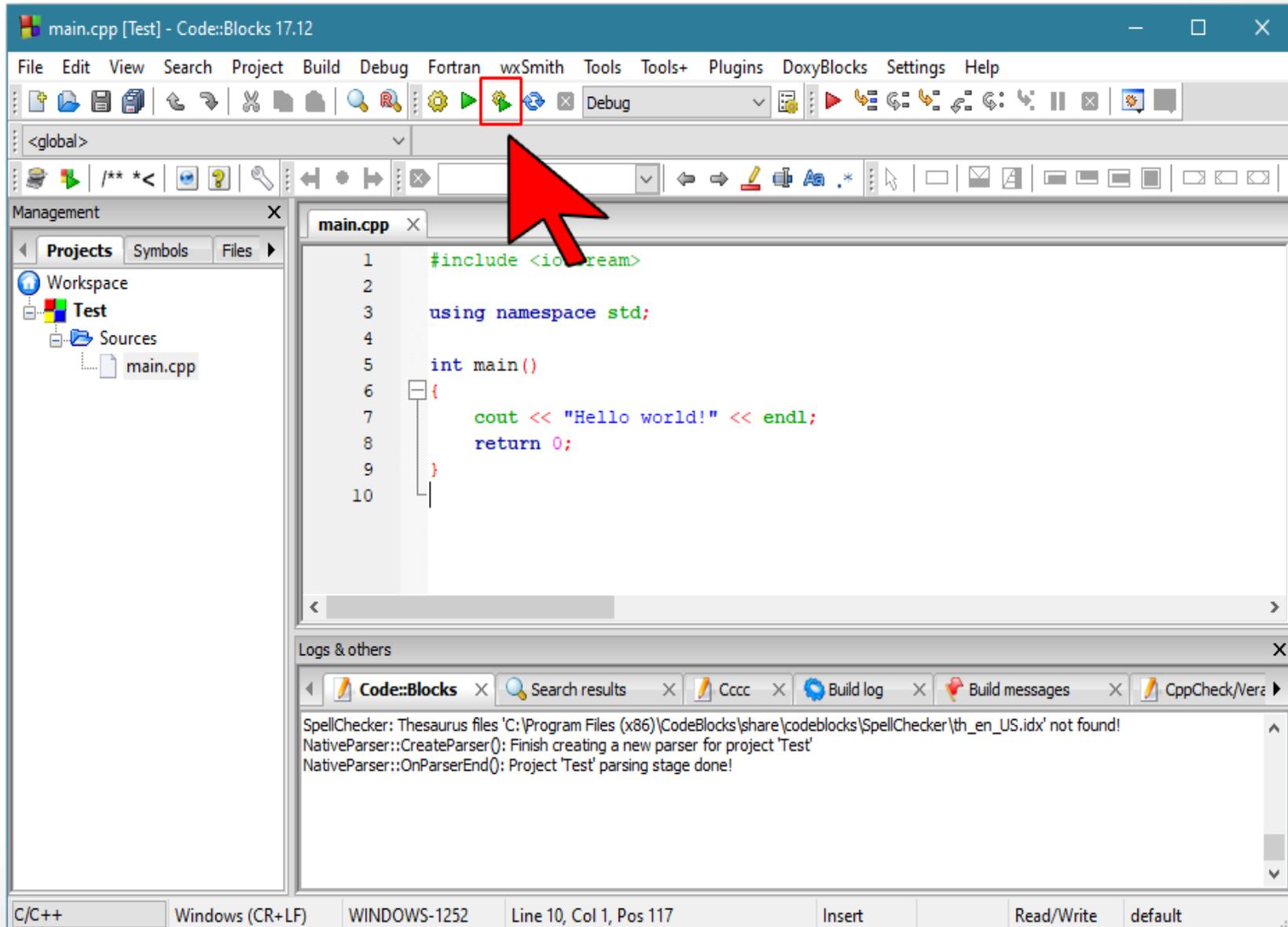
After creating the project successfully, the following window appears



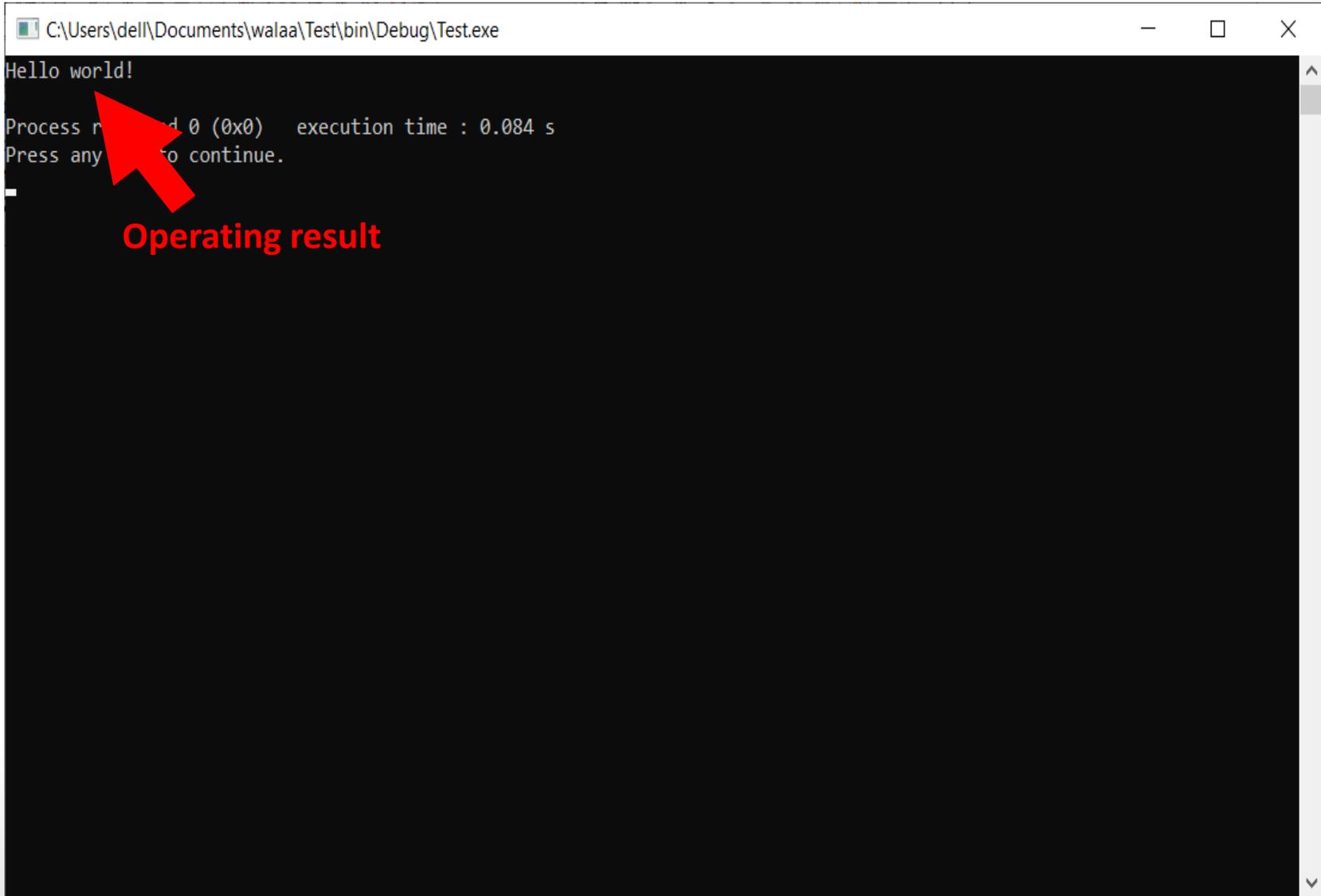
To see the main file in the project that you can write the code into, open the Sources folder and click on the main.cpp file as follows.



To run this program that will only display the phrase "Hello world!" Click on the Build and run button as follows



After running the program, the command prompt will open (which is called CMD or Console) and the results of the operation will appear inside it as follows.



```
C:\Users\dell\Documents\walaa\Test\bin\Debug\Test.exe
Hello world!
Process returned 0 (0x0) execution time : 0.084 s
Press any key to continue.
```

Operating result

Steps to Run the Project Properly

- Before running any project, you must convert the project code to an executable file of its type EXE, and here if the code has any problem, it will appear.
- When you convert all project code to an EXE file you are doing something called Build for the project.
- You do this with just one click and you will learn how to do it with ease

main.cpp [Test] - Code::Blocks 17.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Build Run Build and run

```
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello world!" << endl;
8     return 0;
9 }
10
```

Workspace
Test
Sources
main.cpp

Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Verz

SpellChecker: Thesaurus files 'C:\Program Files (x86)\CodeBlocks\share\codeblocks\SpellChecker\th_en_US.idx' not found!
NativeParser::CreateParser(): Finish creating a new parser for project 'Test'
NativeParser::OnParserEnd(): Project 'Test' parsing stage done!

C/C++ Windows (CR+LF) WINDOWS-1252 Line 10, Col 1, Pos 117 Insert Read/Write default

The image shows the Code::Blocks IDE interface. At the top, the title bar reads "main.cpp [Test] - Code::Blocks 17.12". Below it is a menu bar with options: File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help. A toolbar contains various icons, including a gear (Build), a play button (Run), and a play button with a gear (Build and run). These three icons are highlighted with red boxes, and red lines connect them to three red-bordered boxes containing the text "Build", "Run", and "Build and run" respectively. The main editor window displays a C++ program with line numbers 2 through 10. The code is: `using namespace std;`, `int main()`, `{`, `cout << "Hello world!" << endl;`, `return 0;`, `}`. On the left, the "Management" pane shows a project tree with "Workspace", "Test", and "Sources" containing "main.cpp". At the bottom, the "Logs & others" pane shows several tabs: "Code::Blocks", "Search results", "Cccc", "Build log", "Build messages", and "CppCheck/Verz". The "Build log" tab is active, showing messages from the SpellChecker and NativeParser. The status bar at the very bottom indicates "C/C++", "Windows (CR+LF)", "WINDOWS-1252", "Line 10, Col 1, Pos 117", "Insert", "Read/Write", and "default".

NOTE :

If you only modified the project code and saved it without doing Build - that is, without converting the project code into an executable file - and then run it, you will see when running the result of the last time you did Build and not the result of the code you last saved. So you should do Build or Build and Run

C++ programming

cout

```
cout<< "Text";  
cout<< Number
```

The quotation mark is used when printing text or a letter, and is not used when printing a number.

C++ programming

CIN

```
cin>> a;
```

The program stores the number entered in a

It is preferable to write an explanatory sentence before the input phrase, to tell the user of what he must enter, and more than one variable can be entered in one sentence.

C++ programming

Example :

Write a program in the language of C ++ to show the following phrase:

“Welcome to the second stage students “

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Welcome to Second Stage" << endl;
    return 0;
}
```

C++ programming

Example:

Write a program in the language of C ++ to show the following figure

```
*      #include <iostream>
**
***
****
*****
*****
*****
      int main()
      {
          cout<<" * \n ** \n *** \n**** \n *****";
          return 0;
      }
```

\n : To end the line

C++ programming

Example :

Write a program in the language of C ++ to to enter two numbers and then print their average

```
#include <iostream>

using namespace std;

int main()
{
    int a,b,sum;
    float avg;

    cout<<"Enter first number :";
    cin>>a;
    cout<<"Enter Second number :";
    cin>>b;

    sum=a+b;
    avg= sum/2;

    cout<<"Average :"<<avg;
    return 0;
}
```

C++ programming

Common Errors

- ** The error is in the header files by typing the library name or files incorrectly.
- ** The programmer forgets the start and end brackets of main function.
- ** The programmer forgets the semicolon at the end of each statements.
- ** Input storage at unknown locations.

A graphic design featuring a blue background with a white circle in the center. A dark blue ribbon-like shape overlaps the circle from the right side. The text "Thank you +++" is written in white on the ribbon. The background has a subtle gradient and a white curved shape at the bottom.

Thank you +++