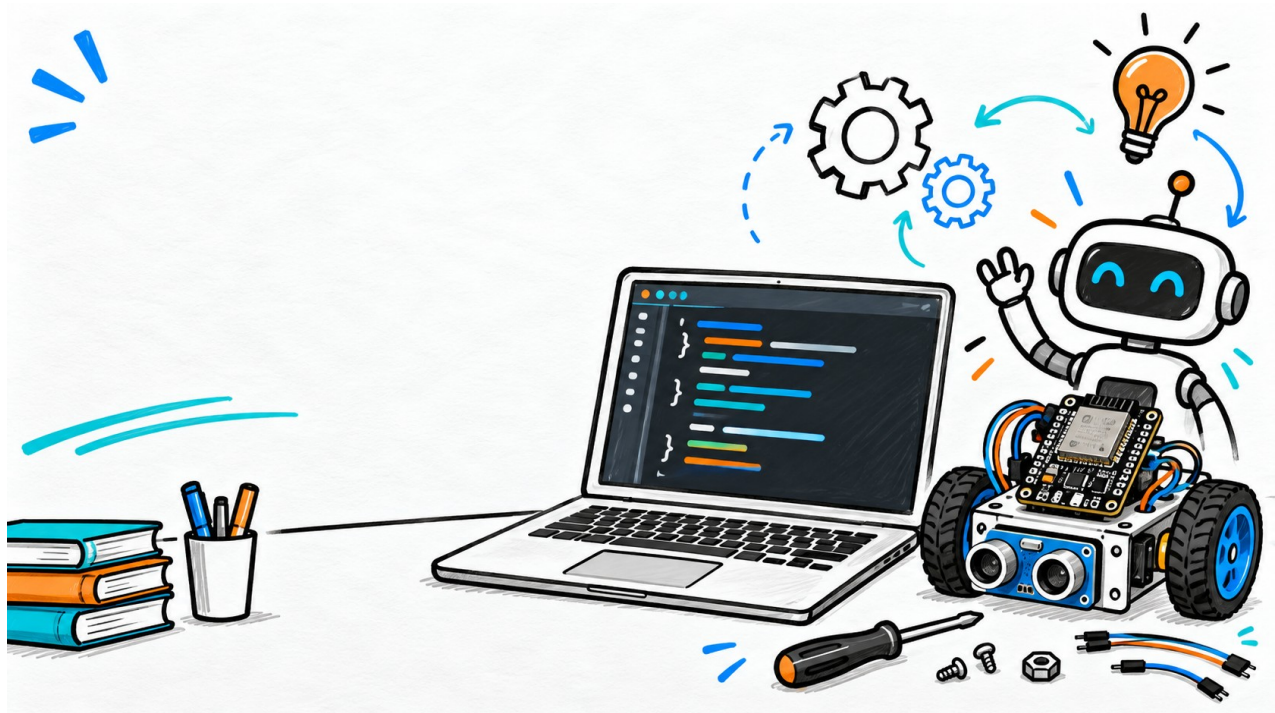


Lesson 1 Complete Teaching Packet

Level 1 C++ Foundations What Is C plus plus



This lesson introduces students to C++ before ESP32 hardware. Students learn that programmers write source code, a compiler translates it, and the computer runs the result. The lesson is intentionally simple so the teacher can measure student confidence, typing accuracy, persistence, and ability to follow the edit, compile, run cycle.

Teacher Quick Start

Item	Details
Time	60 to 75 minutes
Student setup	Windows laptop with VS Code, MSYS2, and g++ installed
Primary outcome	Students compile, run, edit, and explain a Hello World program
Collect	Skill check program or screenshot of terminal output
Best pacing	Slow demo first, then student work time, then short skill check

Learning Targets

- I can explain that C++ is a language for writing computer instructions.
- I can identify source code, compiler, program, compile, and run.
- I can find main, braces, cout, semicolons, comments, and return 0 in a simple program.
- I can compile and run a C++ program from the terminal.
- I can make a small code edit and run the program again.

Materials

- Student Laptop Setup Book
- Lesson 1 Student Workbook
- Lesson 1 Exercises
- Lesson 1 Instructor Answer Key
- Lesson 1 Video Series Script and Artwork Brief
- Student Starter Code folder

Lesson Flow

Time	Teacher Move	Student Move
0 to 5	Ask: What makes a good instruction?	Give examples from school, games, cooking, or robotics.
5 to 15	Explain source code, compiler, program, and the edit compile run cycle.	Fill in key words in the workbook.
15 to 25	Live demo creating main.cpp and typing Hello World.	Watch first without typing unless directed.
25 to 40	Guide students through typing, compiling, and running.	Create file, type code, compile, and run.
40 to 55	Show edits, comments, and one intentional syntax error.	Change output, add comments, fix one error.
55 to 70	Run skill check and circulate.	Build three-line output program.
70 to 75	Exit question: What does the compiler do?	Answer in one or two sentences.

Mini Lesson Teacher Script

Use this as a natural script, not something to read word-for-word.

Today we are starting with plain C++ before using the ESP32. A robot controller is still a computer, and computers follow instructions exactly. The code we type is called source code. The computer

does not run that source code directly. First, a tool called a compiler checks and translates it into a program. Then we run the program and see what it does.

The most important routine today is edit, save, compile, run, and fix. If something breaks, that is normal. The compiler is giving us feedback. Our job is to read the feedback, check the nearby code, and try again.

Starter Program

```
#include <iostream>

using namespace std;

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

Teacher Explanation Notes

Code Part	Say This
#include <iostream>	This gives our program tools for printing text.
using namespace std;	This keeps today's print command shorter.
int main()	This is where the program starts.
{ }	Braces hold the instructions inside main.
cout	This prints text to the terminal.
;	Many C++ statements end with a semicolon.
return 0;	This says the program finished successfully.

Guided Practice Commands

Compile:

```
g++ main.cpp -o hello.exe
```

Run:

```
.\hello.exe
```

Expected output:

```
Hello, world!
```

Student Skill Check

Students create a program that prints this exact output:

C++ Skill Check

I can edit source code.

I can compile and run a program.

The program must include a correct include line, main function, braces, at least one comment, three cout lines, semicolons, and return 0.

Assessment Rubric

Level	Evidence
Ready	Compiles, runs, output matches, and student can point to main, cout, braces, and semicolons.
Almost ready	Compiles after one or two small fixes; student understands the basic workflow.
Needs support	Student cannot yet compile and run without direct step-by-step help.

Common Problems

Problem	Teacher Response
g++ is not recognized	Check setup, PATH, and whether VS Code was reopened after setup.
No output appears	Make sure the student ran the .exe after compiling.
Old output appears	Save, compile again, then run again.
Missing semicolon	Look near the end of the previous statement.
Wrong folder	Open the correct lesson folder or move the terminal there.