

Lesson 1 Video Series Script and Artwork Brief

Level 1 C++ Foundations What Is C plus plus

This production guide supports the first pure C++ lesson. The videos should feel energetic and visual, but still classroom-friendly: clear screen recording, simple vocabulary, and graphics that help students understand the edit, compile, run cycle.

Recommended Video Set

Video	Length	Purpose
1 What Is C plus plus	5 to 7 min	Explain source code, compiler, program, and why robotics students start here.
2 Hello World Walkthrough	8 to 12 min	Show the exact student workflow: create file, type code, compile, run.
3 Fixing Your First Errors	3 to 5 min	Normalize errors and show missing semicolon, quote, and wrong folder fixes.
4 Skill Check Walkthrough	5 to 8 min	Guide students through the three-line skill check without turning it into copy-only work.

Visual Style Direction

Use a polished whiteboard-tech look: clean white background, bold black marker outlines, teal and electric blue accents, small orange highlights for warnings, friendly robotics classroom energy, and layered doodles of code blocks, laptops, gears, wires, and robot parts. Keep text areas open so titles can be added later in the video editor.

- Avoid dark cyberpunk styling for Level 1. It should feel welcoming, not intimidating.
- Use the same title card structure in every clip so students recognize the series.
- Use artwork for title cards, chapter bumpers, and error explanations; use real screen recording for the actual coding.
- Keep on-screen text short: one concept per graphic.

Master Artwork Prompt

Use this as the base prompt for generated title cards and chapter graphics:

A clean high-resolution educational whiteboard illustration for a beginner robotics C++ course, bright white background, bold black marker lines, teal and electric blue accents, small orange highlights, friendly classroom technology feel, laptop showing simple C++ code, small robot

controller board, gears and wires as light doodles, polished but hand-drawn, spacious composition with clear empty area for title text, no readable text, no logos, 16 by 9 aspect ratio.

Video 1 Script What Is C plus plus

Time	On Screen	Narration
0:00	Animated title card: What Is C++?	Welcome to Level 1. Before we touch robot hardware, we need to learn how to give a computer clear instructions.
0:30	Graphic: human code arrow compiler arrow program	C++ is a programming language. We write source code, then a compiler translates that code into a program the computer can run.
1:30	Whiteboard terms: source code, compiler, program	Source code is what you type. The compiler checks it and translates it. The program is what you run.
2:30	Show simple Hello World code, no typing yet	This is a complete C++ program. It looks strange at first, but today we only need to recognize the major parts.
4:00	Graphic: edit, compile, run, fix	Most programming follows a cycle: edit, save, compile, run, and fix. That cycle matters more today than memorizing every detail.
5:30	Closing bumper	In the next video, we will create the file, type the program, and run it ourselves.

Video 2 Script Hello World Walkthrough

Time	On Screen	Narration
0:00	Title card: Hello World Walkthrough	Now we are going to build the first program exactly the way you will do it in class.
0:30	Screen recording: open VS Code folder	Open your Lesson01_HelloWorld folder. Keeping work in the right folder prevents many beginner problems.
1:30	Create main.cpp	Create a new file named main.cpp. The .cpp ending tells us this is a C++ source file.

2:00	Type starter program slowly	Type carefully. C++ cares about punctuation, quotation marks, braces, and semicolons.
5:00	Terminal compile command	Now compile with <code>g++ main.cpp -o hello.exe</code> . If there is no error, the compiler created a program.
6:30	Terminal run command	Run it with dot backslash hello dot exe. The output should say Hello, world.
8:00	Edit cout text	Now change the message, save, compile again, and run again. This proves you are controlling the program.
10:00	Closing checklist	You created source code, compiled it, ran it, and changed it. That is the core workflow.

Starter Code for Video 2

```
#include <iostream>

using namespace std;

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

Video 3 Script Fixing Your First Errors

Error	On Screen	Teaching Point
Missing semicolon	<code>cout << "Hello" << endl</code>	The compiler often points near the problem. Check the end of the line.
Missing quotation mark	<code>cout << "Hello << endl;</code>	Text must begin and end with quotation marks.
Wrong folder	Terminal cannot find main.cpp	Open the correct folder or move the terminal into the project folder.
Old output	Program still prints old message	Save the file, compile again, then run again.

Video 4 Script Skill Check Walkthrough

This video should guide thinking without giving students a pure copy-and-paste answer too early.

Step	On Screen	Narration
1	Show target output only	Start with what the program should print. We need three output lines.
2	Show blank main structure	Build the structure first: include, main, braces, and return.
3	Add one cout line	Once one output line works, repeat the pattern for the next two lines.
4	Add comment	A comment explains the purpose of the program to a human reader.
5	Compile and run	If the output matches, the skill check is complete.

Artwork Prompt Bank

Use	Prompt
Series title card	Whiteboard-tech illustration for Beginner C++ Robotics, laptop with simple code shapes, friendly robot controller, gears and wires, teal and blue accents, open space for title, no readable text, 16:9.
Compiler graphic	Educational whiteboard diagram showing handwritten source code transforming through a compiler machine into a runnable program icon, clean black marker lines, teal arrows, orange caution sparkle, no readable text, 16:9.
Edit compile run cycle	Circular whiteboard workflow illustration with four icons: pencil editing code, save disk, compiler gear, play button, bright classroom style, teal and blue accents, no readable text, 16:9.
Troubleshooting bumper	Friendly whiteboard illustration of a C++ code page with one highlighted semicolon, magnifying glass, small robot helper, orange warning accent, reassuring classroom mood, no readable text, 16:9.
Skill check graphic	Whiteboard illustration of a student laptop, checklist, simple code blocks, robotics team energy, polished hand-drawn style, teal and electric blue accents, no readable text, 16:9.

Production Notes

- Record screen at 1080p or higher.
- Increase VS Code font size before recording.
- Narrate slowly enough that students can pause and follow along.
- Use the same folder and filenames as the student workbook.
- Do not hide mistakes during troubleshooting clips. Show the error, then fix it calmly.
- End each video with a clear next action instead of a long recap.

Teacher Use

- Play Video 1 before the live demo.
- Use Video 2 for absent students or students who need a slower replay.
- Assign Video 3 as the first stop when students hit beginner syntax errors.
- Use Video 4 after students attempt the skill check once.

Video Status

Lesson 1 video: available on this lesson's website page with synchronized captions.

Downloadable transcript: available beside the video on the lesson page.

Classroom use: Play the lesson video, pause for discussion, and then begin the guided student activity.