

Lesson 2 Teacher Lesson Plan

Level 1 C++ Foundations Program Structure and Variables

Lesson 2 builds on Hello World. Students now learn how to read the shape of a program, use comments, create variables, and print variable values with labels. The lesson still stays on laptops only.

Lesson Snapshot

Item	Plan
Audience	Middle and high school robotics students
Length	60 to 75 minutes
Prerequisite	Lesson 1 or equivalent ability to compile and run main.cpp
Big idea	A C++ program is easier to understand when it has clear structure, meaningful variable names, and readable output.
Student product	A robot profile program using string, int, and double variables

Learning Objectives

- Explain why indentation, braces, and comments make code easier to read.
- Create string, int, and double variables.
- Use meaningful variable names with simple naming rules.
- Print labels and variable values with cout.
- Modify starter code without breaking the compile and run cycle.

Vocabulary

Term	Student Friendly Meaning
variable	A named place to store a value.
string	Text, such as a robot name.
int	A whole number.
double	A number that can have a decimal.
initialize	Give a variable its first value.

comment	A note for humans that the compiler ignores.
---------	--

Timing Guide

Time	Activity
0 to 8 min	Review Lesson 1: edit, save, compile, run.
8 to 18 min	Mini lesson: program structure, indentation, comments, and naming.
18 to 30 min	Teacher demo: robot profile variables.
30 to 48 min	Guided practice: students modify starter values and print output.
48 to 65 min	Exercises: add a variable, fix bugs, complete skill check.
65 to 75 min	Exit response and quick share-out.

Starter Program

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string robotName = "PracticeBot";
    int teamNumber = 12345;
    double batteryVoltage = 12.6;
    int motorCount = 4;

    cout << "Robot name: " << robotName << endl;
    cout << "Team number: " << teamNumber << endl;
    cout << "Battery voltage: " << batteryVoltage << " volts" << endl;
    cout << "Motor count: " << motorCount << endl;

    return 0;
}
```

Teaching Notes

- Do not over-explain memory. For now, a variable is simply a named value.
- Keep naming rules practical: no spaces, begin with a letter, use meaningful names.
- Use robot-themed values so variables feel connected to later robotics work.

- Treat formatting as part of communication, not decoration.

Skill Check Criteria

Skill	Evidence
Variables	Student creates at least one string, int, and double.
Output	Student prints labels and values clearly.
Naming	Variable names are meaningful and follow simple rules.
Structure	Code has indentation, braces, semicolons, and at least one comment.
Workflow	Student can save, compile, run, and fix one simple error.