Introduction to C++

Ryan Baker

January 2, 2025

Contents

1 Course Introduction

1.1 Overview of Lecture Series

2 Features of C++

2.1 Evolution of C++

C++ was developed in 1979 by Bjarne Stroustrup as a simple extension of C. Since then, it has evolved into a modern multi-paradigm language with major updates every three years (C++26 is on its way). Each update introduces features for better performance, safety, flexibility, and developer experience.

2.2 The C++ Philosophy

C++ is a sharp tool. It prioritizes manual control over all else, allowing for direct memory manipulation and fine-grained resource management. The philosophy is to give the developer the tools for flexibility and performance, but with the responsibility to manage complexity.

2.3 C++ vs. Other Languages

Compiled vs. Interpreted C++ is a *compiled* language, meaning the source code is translated into machine code before it is executed. This allows for a faster run time and more control over hardware aspects. This contrasts with *interpreted* languages, like Python, translated and executed line by line. Interpreted languages tend to be quicker to develop and easier to use.

Strongly Typed Strongly typed languages, such as C++, require the explicit specification of datatypes and enforce type assignments at compile and run time. This makes it harder to make type mistakes and promotes code stability.

Multi-Paridigm A programming paradigm is a pro

3 Environment Setup

3.1 Tools Required

3.1.1 Text Editor

3.1.2 Compiler

3.2 “Hello, World!” Example

4 Basic Syntax and Structure

4.1 Basic Structure of a C++ Program

4.1.1 `int main()`

4.2 Foundational Concepts

4.2.1 Semicolons, `/* comments */`, and Whitespace

4.2.2 Line-by-Line Execution

4.3 Input and Output

5 Datatypes and Variables

5.1 Primitive Types

5.1.1 `int`, `char`, `bool`, `float`, `void`

5.1.2 `sizeof` Operator

5.2 Declaration and Definition

5.2.1 Assignment Operator `=`

5.2.2 Brace Initialization `{}`

5.3 Arithmetic Operators