



Week 2: C++ Fundamentals - Variables, Input, and Arithmetic

Welcome to week 2 of Programming Fundamentals (C++)! This presentation will cover variable declarations, data types, user input, and arithmetic operators. We'll lay the groundwork for more complex C++ programming concepts.



Lecture 3: Variable Declaration and Memory Concepts

What is a variable?

A variable is a named storage location in memory. Think of it like a labeled box. It holds data during program execution.

Analogy

You can also think of it as a labeled box to hold data.

Why use variables?

We use variables to store and manipulate data during program execution.

Data Types: Integer Variables

``int``

Stores whole numbers like -10, 0, or 25. Requires typically 4 bytes of memory. Range: Approximately -2 billion to +2 billion (platform-specific).

Example

``int age = 30;`` Common uses: Counters, indices, quantities.

Use cases

Common uses: Counters, indices, quantities.

Data Types: Floating-Point Variables

``float``

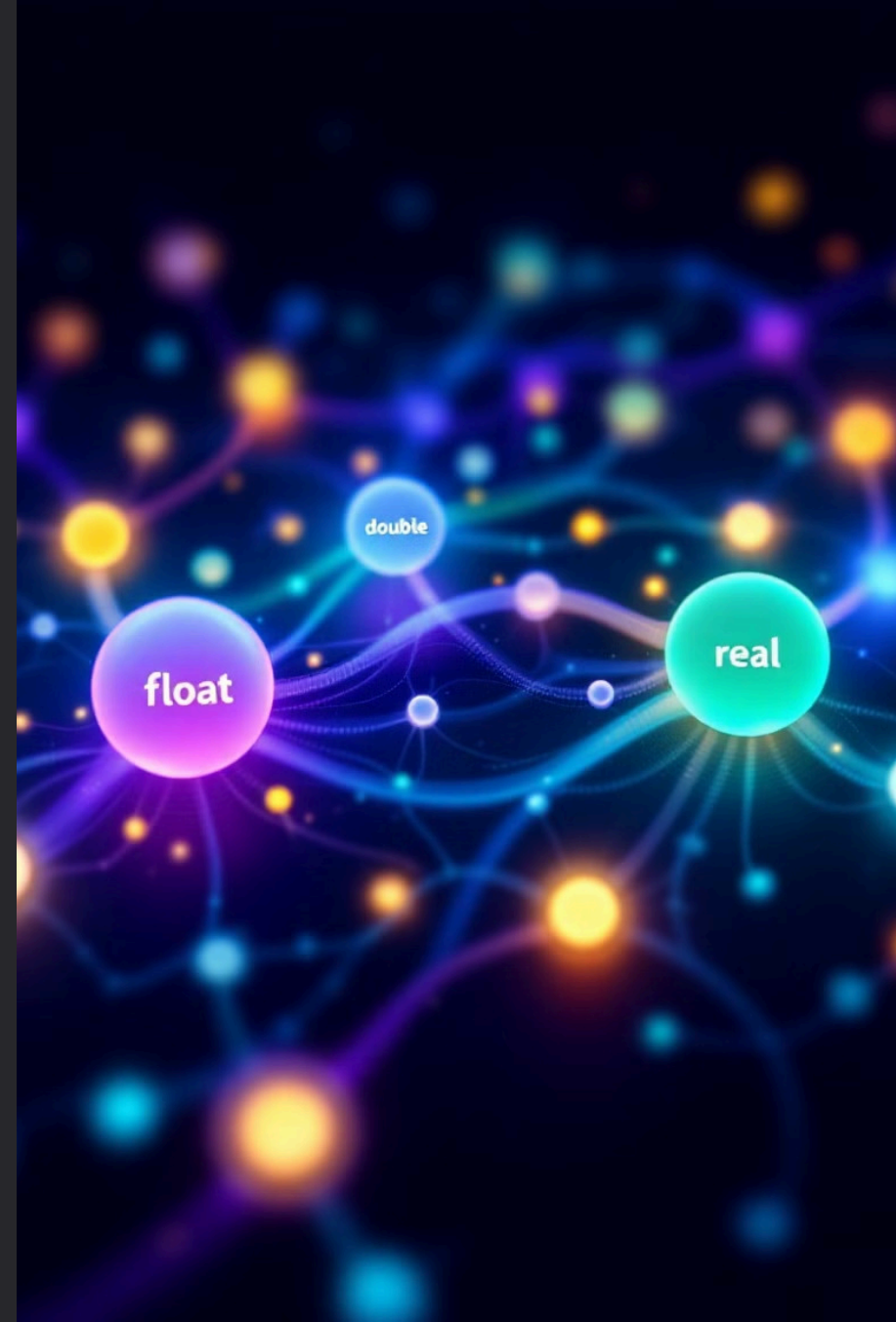
Stores single-precision numbers (e.g., 3.14, -2.71). It takes 4 bytes, with ~7 digits of precision.

``double``

Stores double-precision numbers, using 8 bytes and offering ~15-16 digits of precision.

Example

``double price = 99.99;`` Good for measurements, calculations, and monetary values.



Variable Initialization



Definition

Assigning an initial value to a variable when it's declared.



Why Initialize?

Initialization avoids undefined behavior and garbage values.



Methods

Direct: `int count = 0;` Copy: `int count = 0;` Uniform (C++11):
`int count{0};`



Taking Input from User Using `cin`

1

``cin``

Standard input stream object in C++.

2

Purpose

Reads data from the console (keyboard).

3

Syntax

``cin >> variable_name;``

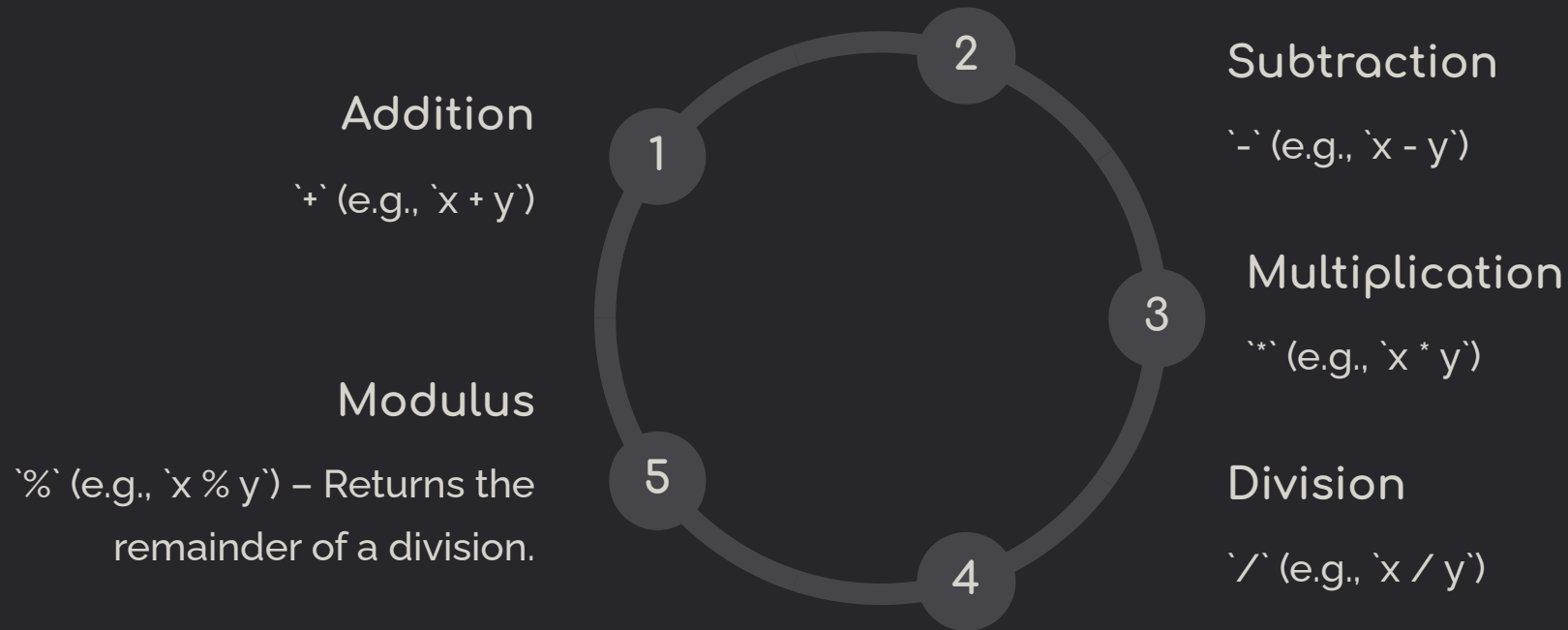
4

Example

``int age; cout << "Enter your age: "; cin >> age;``



Lecture 4: Arithmetic Operators



Arithmetic Expressions

1

Definition

Combinations of variables, constants, and operators.

2

Order of Operations

Parentheses, multiplication/division/modulus, addition/subtraction

3

Example

``result = (a + b) * c / d;``

Integer Division and Modulus

Integer Division

If both operands are integers, the result is an integer (truncation). For example, `5 / 2` evaluates to `2`.

Modulus

Modulus returns the remainder of integer division. For example, `5 % 2` evaluates to `1`.

Recap and Next Steps

Key takeaways

Today we went over variables, data types, initialization, `cin`, and arithmetic operators.

Practice

Write programs using variables, input, and arithmetic operations.

Next Week

Control flow (if statements, loops) are next up!

