The Components of C++: A Look Under the Hood

by Atomic Badger
3-5-2025

Parts of C++:

C++ can be defined as a totality by the following 8 categories of parts:

- *entities*
- expressions
- statements
- declarations and definitions
- types
- scope and duration of storage
- templates
- preprocessor directives

Everything else traces back to how one of these parts is used.

1. **Entities:**

An entity is a fundamental building block. An entity may include:

- variables
- functions
- classes
- objects
- namespaces

2. **Expressions:**

An expression is something that produces a value, Expressions may cause side effects.

Expressions may include:

- literals
- variable references
- function calls
- operator expressions
- constant expressions
- integral exoressions
- floating-point expressions
- relational expressions
- logical expressions
- bitwise expressions
- assignment expressions
- arithmetic expressions

- pointer expressions
- conditional expressions
- comma expressions
- Lambda expressions
- function call expressions
- new and delete expressions

Each type of expression serves a specific role in C++ and contributes to C++'s overall logic and program flow.

3. **Statements:**

A statement is a complete interaction that performs an action. A statement may or may not include expressions in its composition.

Statements may be one or more of the following types:

- expression statements
- declaration statements (which may declare constants, variables, and functions)
- control flow statements - decisions (if – if else – switch)
- iteration statements – loops (while – do/while – for)
- jump statements (break, continue, return, goto)
- block statements
- exception handling (try – catch)

4. **Declarations and Definitions:**

Declarations are a way of introducing entities.
Definitions allocate memory space for entities.

Definitions and declarations may include:

- variable declarations
- variable definitions
- class declarations
- class definitions

5. **Types in C++**

- primitive types
- compound types
- user-defined types

6. **Scope and Duration in Storage:**

- local scope
- global scope
- namespace scope
- storage duration

7. **C++ Templates:**

Templates are a programming feature that allows for writing flexible and reusable code.

8. **Preprocessor Directives:**

Preprocessor directives are instructions that modify program code before compiling.

Examples of preprocessor directives:

- #include
- #define
- #IFDEF