Introduction to C++

What c++?

C++ is a programming language just like Java. Unlike Java though, C++ is extremely flexible, allowing users to have complete control of their machine's resources. It provides low-level memory manipulation. All these factors allow you to make your programs run as fast and efficiently as possible.

- For these reasons, C++ is used in fields like videogame development, finance, and systems programming!

Java vs C++

What are the differences & similarities between Java and c++?

Similarities

- Syntax: Both languages share similar systanx like {} () and ;
- OPP: Both are object-oriented programming languages. Concepts like classes objects, inheritance, polymorphism, and encapsulation.
- Standard libraries: Both languages have extensive Standar libraries that provide built in functionality for tasks like Data structures, file handling and networking.
- platform independence: Java is platform independent achived through the Java virtual machine. C++ can be made platform independent by careful coding.
- Memory management: Both languages allow manual mem management, though Java handles most of it automatically through the Garbage collection, while C++ requrires explicit memory management.

Differences

Java

- Runs on the JVM, making

C++

- platform dependent; code must be recompiled

platform independent

for different platforms.

- Memory management is handled by the garbage collection.

- Memory management is handled by the developer.

- performance is slightly slower due to the JVM overhead.

- Faster and more efficient, as it compiles directly into machine code.

- Does not support multiple inheratance, uses interfaces instead.

- supports multiple inherantances.

- uses generics, which are erased at runtime.

- uses templates, which are resolve at compile time.

Compilation:

Compilation:

Java code

↓

Bytecode

↓

runs on JVM

C++ code

↓

Machine code

How C++ Compiler works

Unlike Java, C++ compiles your C++ code directly into machine code. This means no interpreter is required like the JVM for Java.

Here's how C++ is compiled:

1. Source code: you write C++ code in text file or editor

2. Preprocessing: The compiler first runs a preprocessor that handles things

like #include (to add other files) and #define (to replace macros)

3. Compilation: The compiler translates your C++ code into machine code [Asembly code] specific to your computer's processor. This is a low-level language your CPU can understand.

4. Linking: if your program uses external libraries ar multiple files, the linker combines all the compiled code into a single exe file.

5. Execution: you run the exe and your program does what it's suppose to do.

What is namespace?

Namespace is a way to organize code and prevent naming conflicts. It acts like a container for identifiers, (such as variables, functions, and classes) to group them the same name.

- Imagine you have two libraries, and both define a function called print();.

Without name spaces, the compiler wouldn't know which print() youre reffering to.