



```
1
1 #include <iostream>
2 using namespace std;
3
4 namespace namespace1;
5 namespace namespace2 {
6
7     return + first only;
8     namespace class: {
9         return full super (ly);
10        //perpices for the "addrection" class);
11    }
12    return 0;
13    return {
14        (iam en++);
15        poll class /step fin);
16    }
17 }
18
19 ; return liam deaps * late;
20
21 }
```

Programming Fundamentals: C++

Welcome to Programming Fundamentals! This presentation covers the history of C++, basic program structure, and essential elements like directives and comments. We'll also explore output methods and manipulators. Let's begin our journey into the world of C++.

The History of C++

C++ evolved from "C with Classes" in 1983. Bjarne Stroustrup sought efficiency and flexibility. The language added object-oriented features. It had its first ISO standard in 1998. C++ continues to evolve with new standards.

1

1979

C with Classes

2

1983

Becomes C++

3

1998

C++98 Standard

4

2011 Onward

Modern Updates



Compilers vs. Interpreters

C++ employs compilers to translate code into machine code before execution. This leads to faster execution speeds. Interpreters translate and execute code line by line. C++ is typically a compiled language.

Compilers

- Translates the entire code at once
- Faster execution
- Examples: g++, Clang

Interpreters

- Translates line by line
- Easier debugging

Basic C++ Program Structure

Let's explore the structure of a simple C++ program. This structure includes directives, comments, output using "cout", escape sequences, setw, endl, and manipulators. Understanding this structure is crucial for writing C++ code.

```
#include iostream
<int main() >
    int main(>;>
std-coutHello-World! endl:
    strings
return 0:
```

```
#include <iostream>

int main() {
    // This is a comment
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```