lecture2

TL;DR:This post explores the fundamental distinctions between C++ and Python, focusing on memory management, type systems, and the beauty of modern C++11 features like type deduction.
Notes inspired by Stanford CS106L:Standard C++ Programming

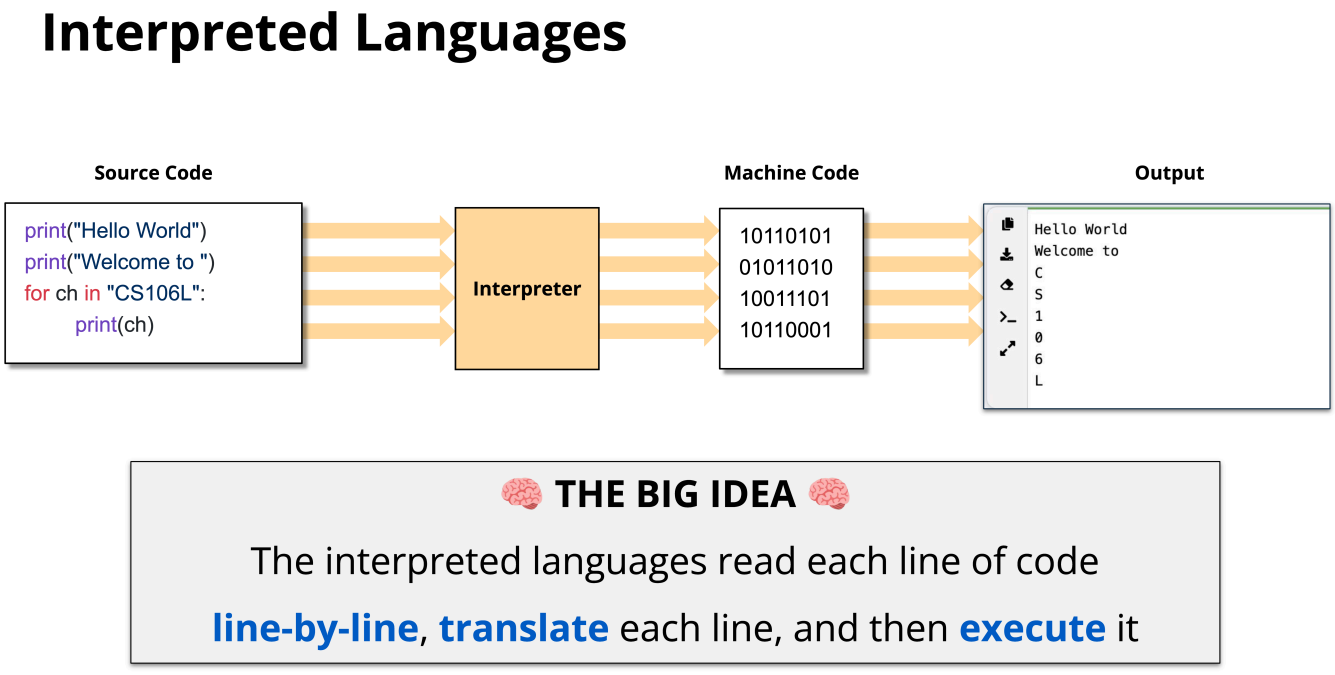
section1:Language Fundamentals

C++ is a **compiled** and **statically typed** language,whereas Python is **interpreted** and **dynamically typed**.

1.interpreted languages(python) and compiled languages(c++)

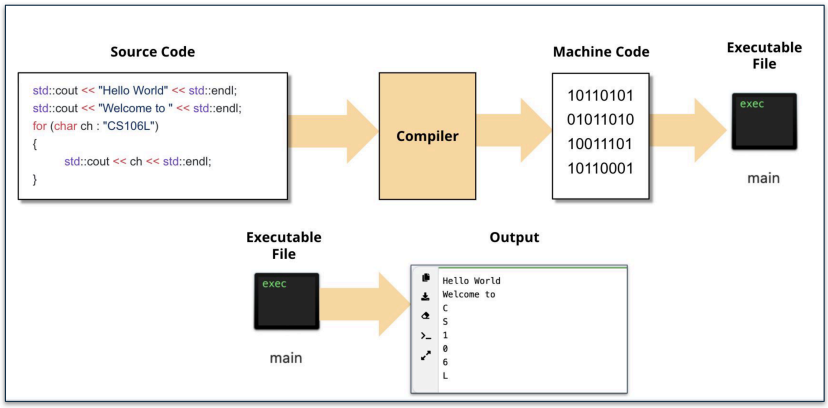
2025Fall-02-TypesAndStructs, p.8

> Compiler VS Interpreter



distinction:

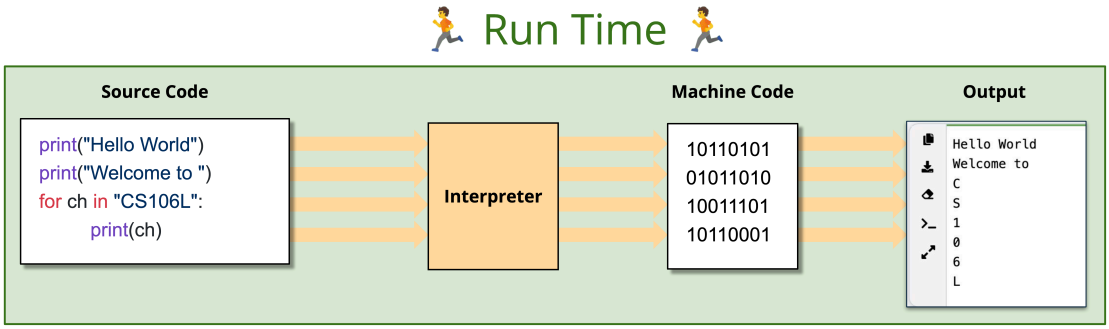
Compiled Languages



🧠 THE BIG IDEA 🧠

The compiler translates the **ENTIRE** program, packages it into an **executable file**, and then **executes** it

Interpreted Languages: Compile Time V.S. Run Time



🧠 THE BIG IDEA 🧠

Interpreted languages all run in **run time**!

Compiled Languages run in first **compile time** then **run time**

2.types:

basic types:int,double,string,bool,size_t

> 2025Fall-02-TypesAndStructs, p.38
>
> C++ is a statically typed language

c++ is a statically typed language.

difference:python is a dynamic typing language.

Advantages of Static Typing:
- **Performance*:Highly efficent as types are resolved at compile-time.**
- Safety**:Better error checking:the compiler catches type mismatches before the program runs.**
- Readability**:Code is more explicit(self-documenting)

> 2025Fall-02-TypesAndStructs, p.50
>
> C++ is a compiled, statically typed language

Takeaway:c++ is a compiled and statically typed language.

Section2:Strcuts and Templates

structs:

A struct bundles named variables into a new type.This is the foundation of creating custom data structures.

we use `struct` to group related attributes into a single,cohesive entity.

pair

pair:is a template

```
template<typename T1,typename T2>
struct pair{
    T1 first
    T2 second
};
std::pair<std::string,int>
```

Section3:Modern C++ Features

some knowledge:

```
<string> std::string
<utility> std::pair
<iostream> std::cout
```

tips:
If we write `using namespace std` ,it is a bad style as it can introduce ambiguity
for example:string:if we introduce `using namespace std` ,it will cluter the global namespace,if we name a varibles as 'string',then we will get an error,because we have introduced the 'string'.

```
using namespace std;
int string=5;
string s;-->this is an error from using namespace std
```

using:**TypeAliases**:Use `using` instead of `typeof` for better readability

```
using zeros=std::pair<double,double>
```

auto:**Type Deduction**:The compiler **deduces** the type of a variable from its **initializer**
auto is still statically typed
in c++11,the complier can deduces the types of x from its initializer.

Recap

- C++ is a compiled, statically typed language
- Structs bundle data together into a single object
- **std::pair** is a general purpose struct with two fields
- `#include` from the C++ Standard Library to use built-in types
  - And use the std:: prefix too!
- Quality of life features to improve your code
  - using creates type aliases
  - auto infers the type of a variable