Transition to C++

1. C++ → Bjarne Stroustrup, 1980s

Java → Interpreted ?? / Compiled ?? +

Interpreted

translated one statement at a time → machine code

Compiled

saw entire program at once → translate all of it to machine code

Java program —compiled→ Byte code —interpreted→ JVM

C++ → compiled language

Python → interpreted language

```
a = 10
if (a == 10):
    print ("Hello")
```

a = "Riya"
a = 'c'

Static & Dynamic typing

Java, C++

int val = 10 ✗✗
char val
String val
(statically typed languages)

Python

a = 10
a = "Akhil"
a = 'c'

	Java	C++
1. Comments	// /* */	// /* */
2. classes		C++ → int main() { } (Hello)
4. Libraries	import	

5. System.out.println ("Hello") → cout << "Hello" >
scn.nextInt (___) → cin <<

Data Types and operators

1. short
2. int
3. long
4. float
5. double
6. char

7. Boolean → bool
0 → false
non-zero → true

8. byte ✗✗

Typecasting → same

== != < > >= <=

operator overloading → to provide the operators with a special meaning for a data-type

2. pointers

.
?:
::

Strings → C++

2 types of strings

1. char array → single char
"Riya" → | R | i | y | a | '\0' |

2. String class → string val = "class"
Java: String str = "Riya"

s.size();
s.length()
s.empty()
s.find(
s.substr(i, n) → substring of length (n) starting at position (i)

"abc" + "def" → abcdef

C++ strings are mutable

s[index] → ✓
s[index] = new char ✓

s.insert(index, t)
"abc" → 1 → de → adebc

Java	C++
final	→ const

long → double → Big Integer

long long → long int →
long long int → (-2^{63}) to $(2^{63})-1$

short int

```
class paper {
  public:
  [ ]
  private:
  [ ]
}
```

class paper()
private
public
private

break, continue

Arrays

Java
Array / Arraylist
size
homogeneous
int days[];
int[] days;
int[] days = new int[5];

C++
array / vector
size
homogeneous
int days[20];

Python
list
✗ size
✓ heterogeneous
list = ["apple", 2, "mango", 17, 18]

int days[] = {31, 28, 31, 22, ...}
double
char

int val = 20;
int val(20);

char greeting[] = "Hello";
string message[20];