

Life Cycle - C/C++

(User-Side)

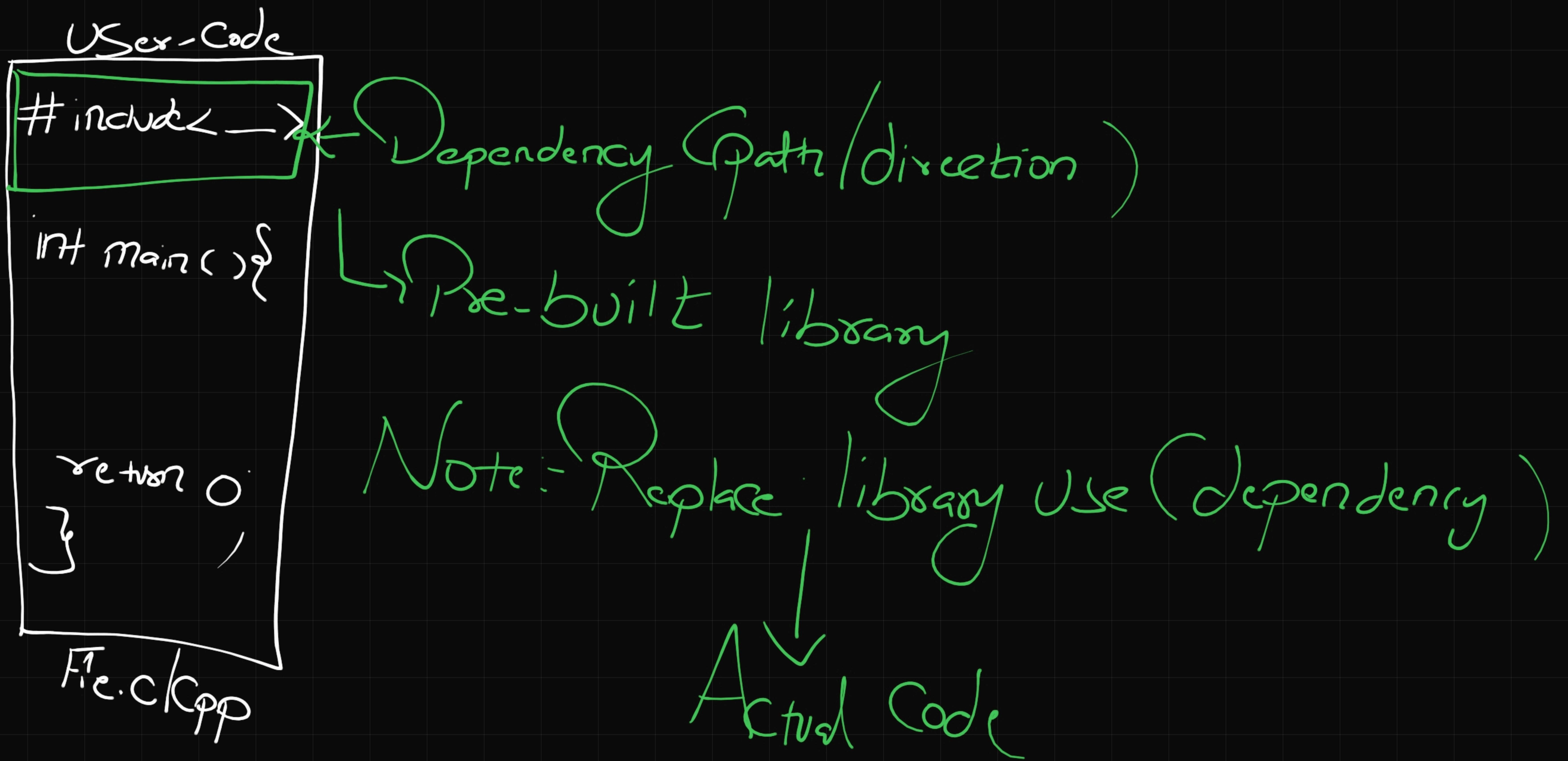
```
#include <iostream>
Void
int main()
{
    std::cout
}
```

File. C

Notes- int main() {
 } return 0;

- i) Not Compulsory to Mention
 - Return type (int) + Return Keyword
- ii) For "OS" to know
 - File successfully execute
 - Error (generate)
- iii) "Void" & Return type
 - Return (Exclude)

ii) Pre-Processor :-



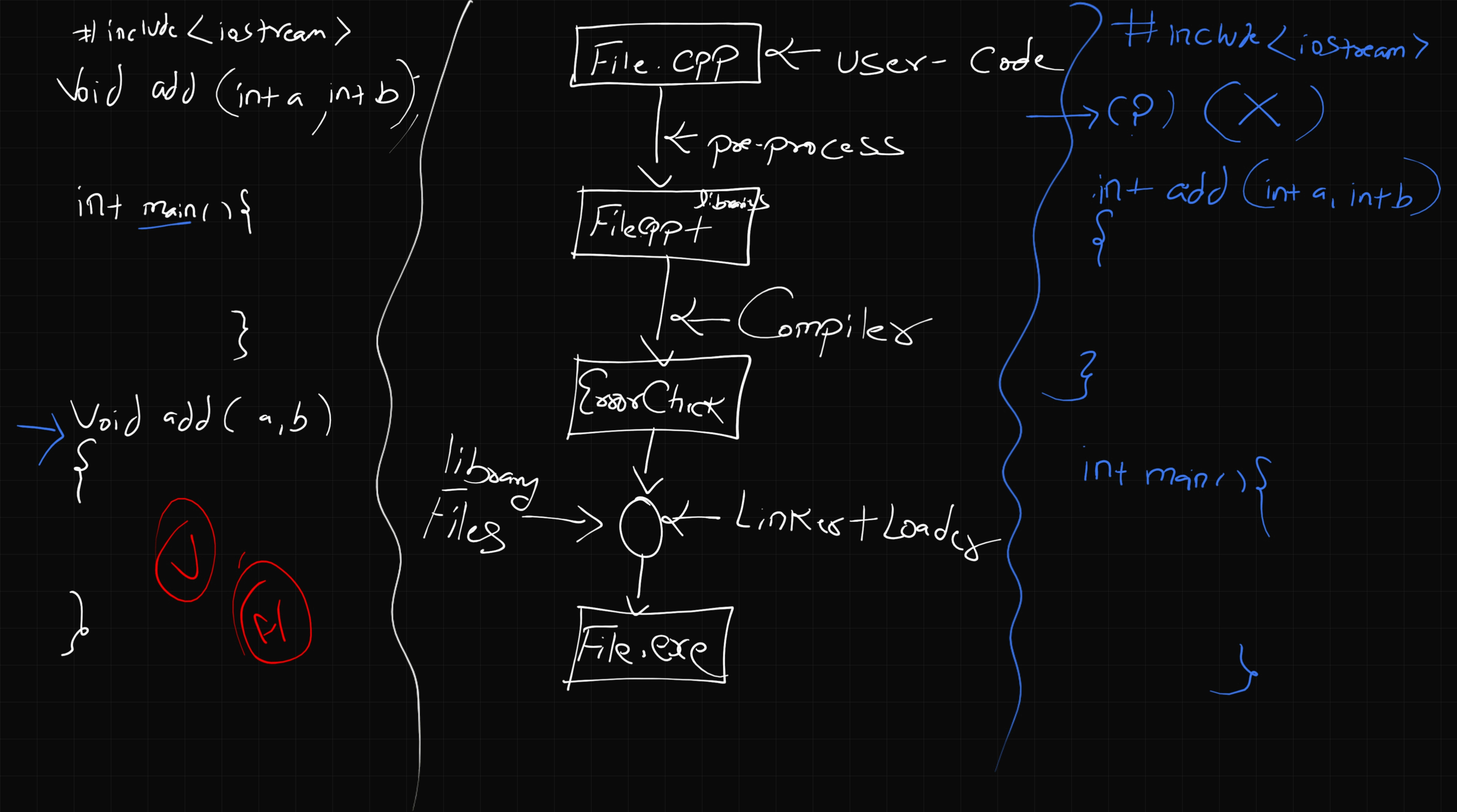
③ Compilation - "Before Creating any machine code it
will check the Syntax-Errors"

→ No-Errors

→ Errors

④ Loader + Linker - Convert it into MachineCode

↳ File.exe



Setup :-

- i) MinGW C/C++ extension

- ii) VSCode -> Extension

- iii) Setup + Environment Variable

- iv) Cmd -> _____

- v) VSCode -> Execute(✓)

C-language

```
#include<stdio.h>
int main() {
    printf(" ");
    scanf(" ");
}
```

C++-language

```
#include<iostream>
#include<cstdio>
int main() {
    (I/O)->cin >> cout << (I/O)
    printf(" ");
    scanf(" ");
}
return 0;
```

Input / Output

printf()

scanf()

printf()

scanf()

Pointf()]
Or
Scanf()] → Justify() -

Int Count = 10, ↓ Time waste

Cin

Cout << Count << endl;

Note: Human efforts

ii Time-efficiency

> Cin

> Cout

> Pointf

> Scanf

Time-Efficiency: $\text{printf} + \text{scanf} > \text{cin} + \text{cout}$

↓
Explicitity

Datatype

↳ $\text{printf}(\text{"Statement"} \underline{\underline{\%d}}, \underline{\underline{\text{Count}}})$

↓
" "

$\text{cout} \ll \boxed{\text{Count}}$ endl;

Datatype(?)

↳ Print

C++ (Basic Programming) - 09 Data Types

② Variables (to store/hold the data)

③ Printing → Cin + Cout

Eg:- ~~int~~ using namespace std;
~~int~~ main() {
 Variable
 Std::cout << "Data is: " << cin << endl;
}

