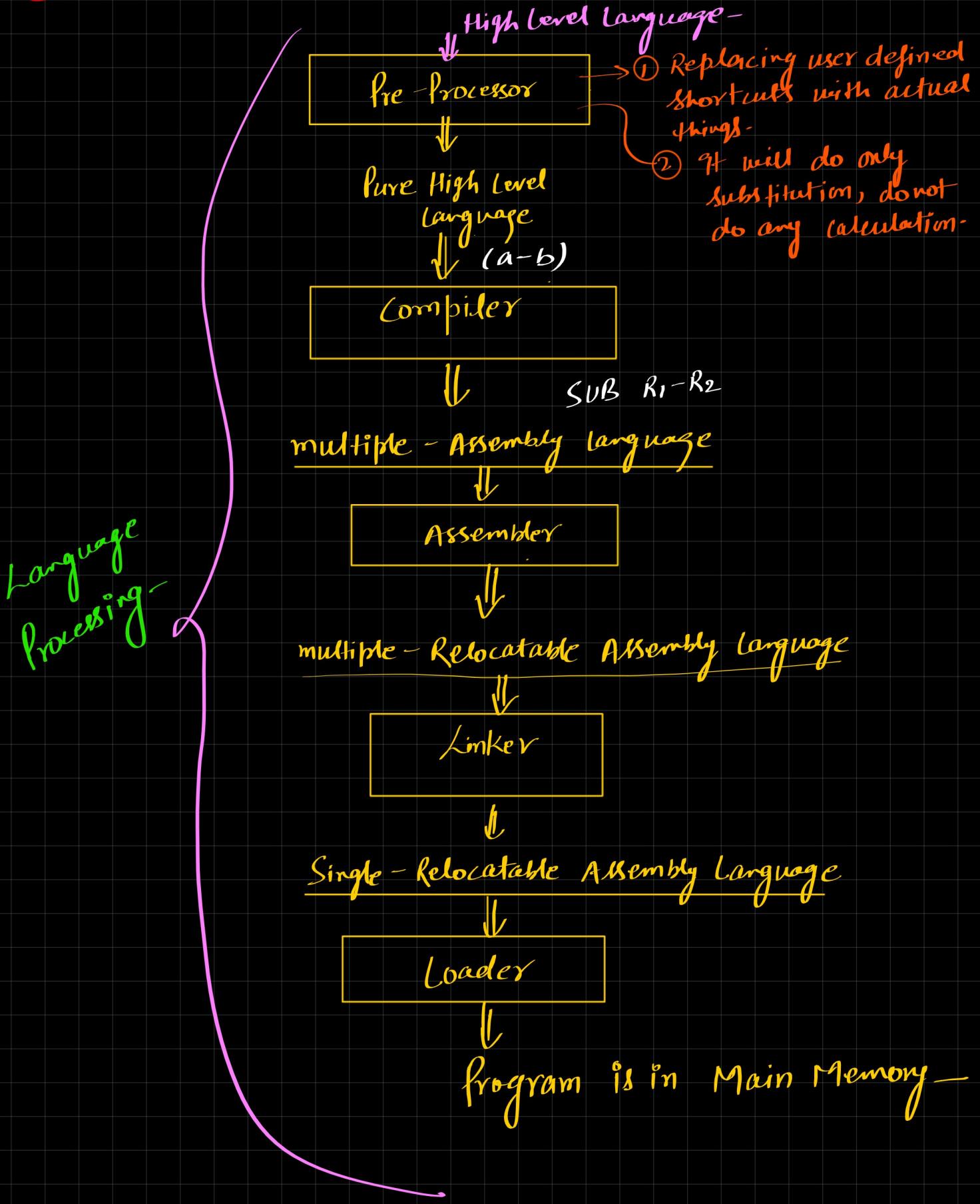


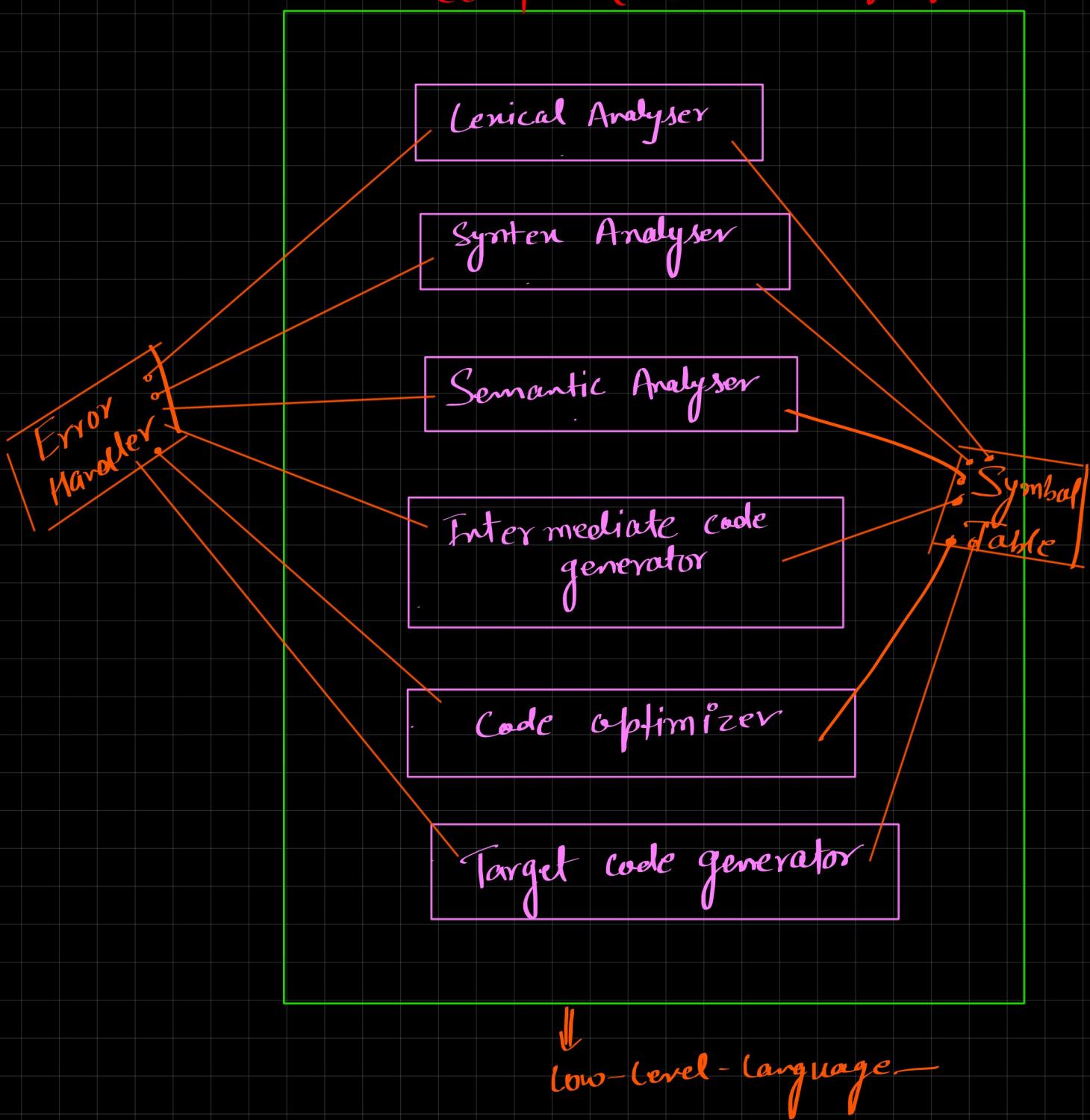
# ④ Introduction to Compiler →



• Assembler  
• Linker  
• Loader }      Cousins of Compiler

④ Compiler :-

Compiler (High-level language)



$$\text{Ex. } \boxed{x = a + b * 60}$$

↓  
Lexical Analyser  
↓

S.NO	V. Name	V. Type
1	x	float
2	a	float
3	b	float

$x$  - identifier -  $\langle \text{id}, 1 \rangle$   
 $=$  - Assignment operator  
 $a$  - identifier -  $\langle \text{id}, 2 \rangle$   
 $+$  - Binary Addition  
 $b$  - identifier -  $\langle \text{id}, 3 \rangle$   
 $*$  - Binary Multiplication  
 $60$  - int constant

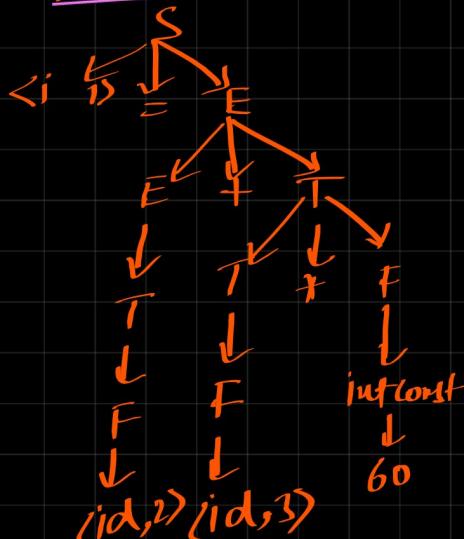
$$\downarrow$$

$$\langle \text{id}, 1 \rangle = \langle \text{id}, 2 \rangle + \langle \text{id}, 3 \rangle * 60$$

↓  
Syntax Analyser

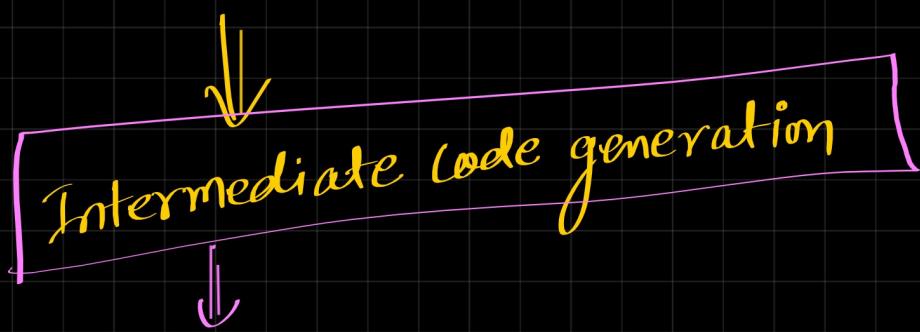
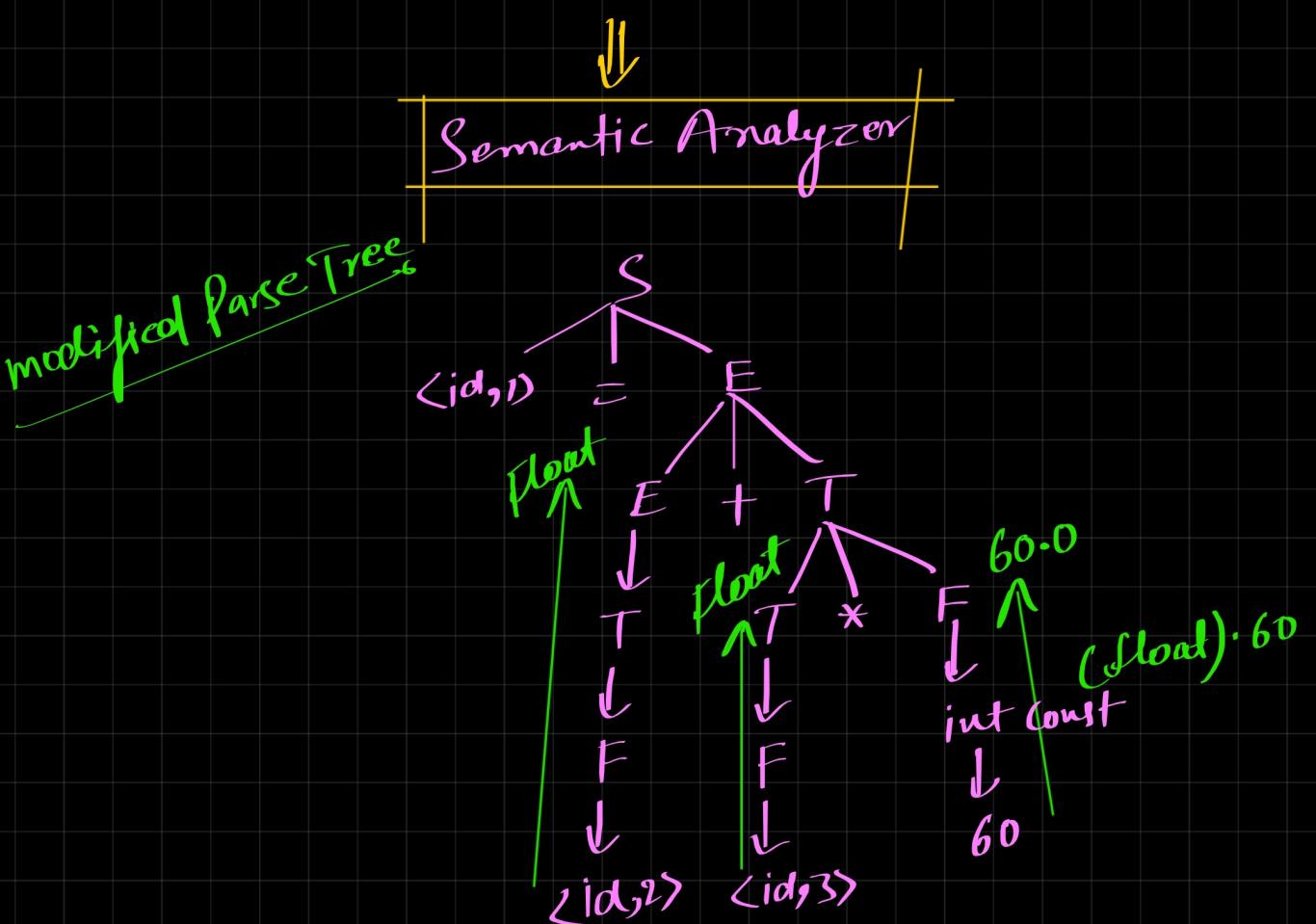
$S \rightarrow \text{id} = E$   
 $E \rightarrow E + T \mid T$   
 $T \rightarrow T * F \mid F$   
 $F \rightarrow \text{id} \mid \text{int const}$

• Parse Tree •



W.  
NOTE

Q:- Symbol table is created by Semantic Analyser at the time of scanning declaration (by Len Analyser) Len inform to Semantic to do that.



$t_1 = b * 60.0$

$t_2 = a + t_1$

$x = t_2$



$t_1 = b * 60.0$

$x = a + t_1$

MUL  
ADD  
Store

Ex.

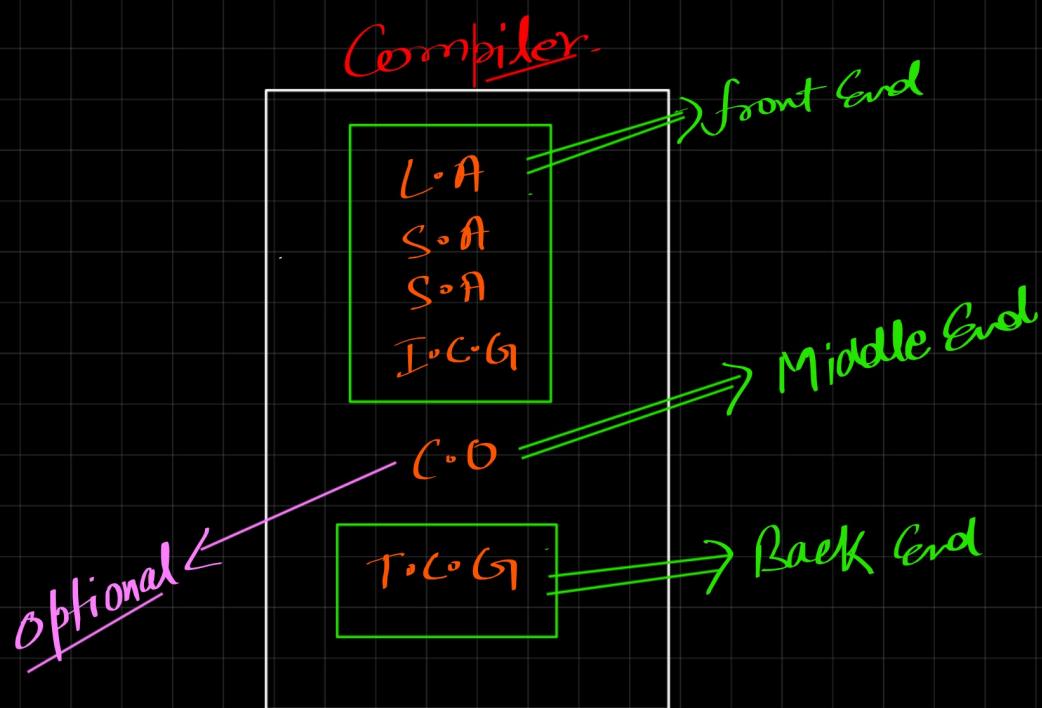
```
main()
{
    a = b + c; // Semantic error
}
```

Ex.

```
main()
{
    int a, b, c; // Syntax error
    a = b + c;
}
```

→ C Compiler → Machine dependent { front end }  
                  ↳ Target code                   { and  
  Back end }

→ Java Compiler → Intermediate code { front End }  
  or Byte code



- If total compilation is done at a time, then it is called Single pass compiler.  
Advantage — less time  
drawback — More space required.

- If total compilation is done in multiple passes then it is called Multi-pass Compiler.

drawback — More time  
Advantage — Less Space.