Page No.
Date 23 01 23

Compiler Design

* Textbook :→
① "principles of compiler design" by Aho and Ullman.
② "Compiler Design" by O.G Kakde.

Compiler

Source Program → Translator (compiler) → Target program

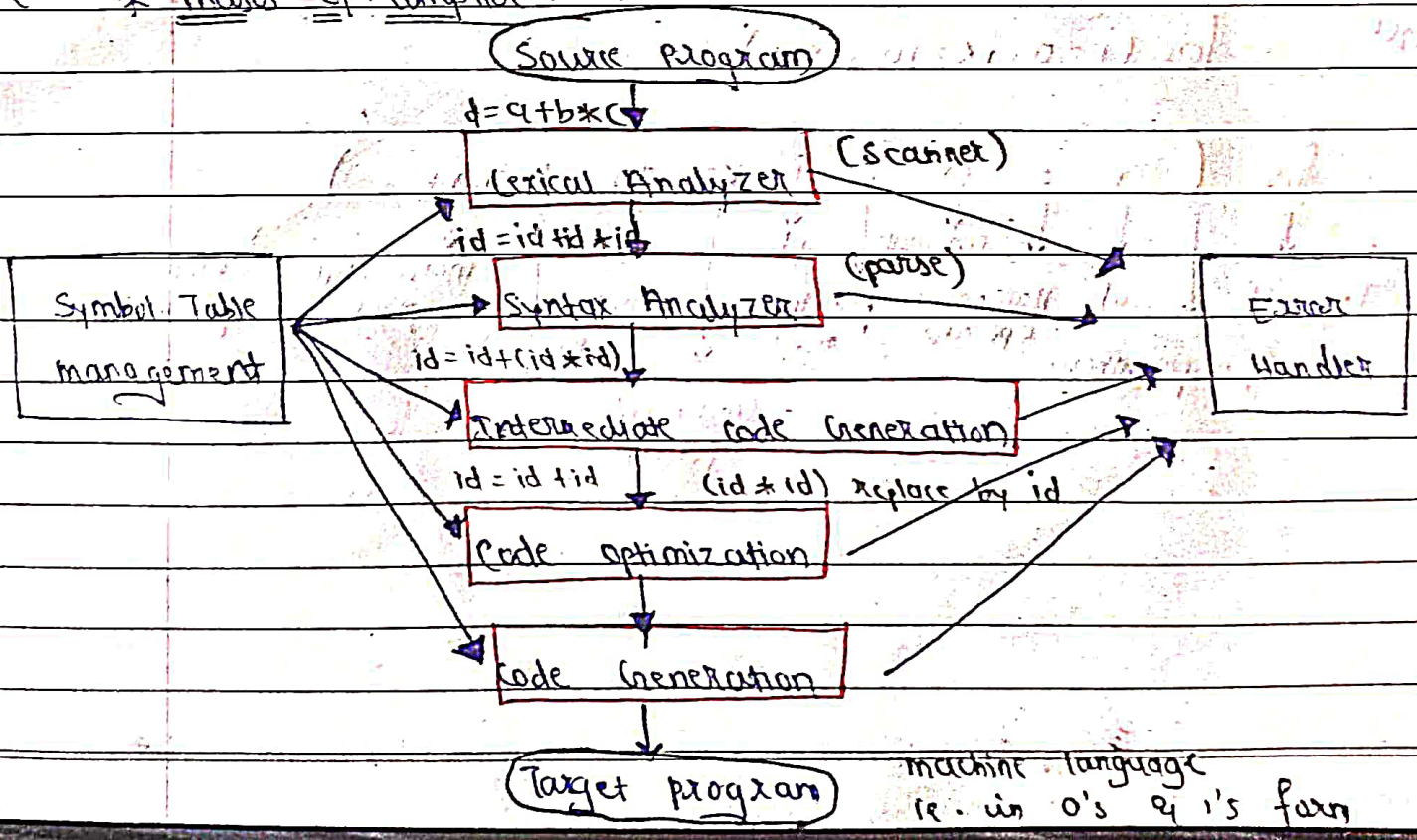
(High level languages)

(machine code) → 0 & 1 form

Example of Translators → ① Interpreter
② Assembler → ADD 1, 2 / MOV A, B

(Imp (13 marks)

* Phases of compiler :→



machine language
i.e. in 0's & 1's form

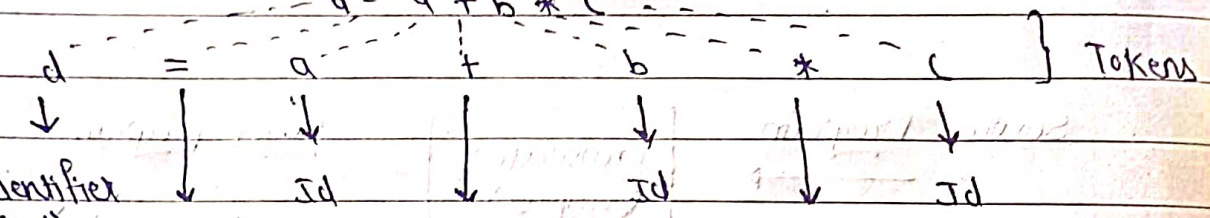
If we have an expression

d = a + b * c

then

① Lexical Analyzer :→

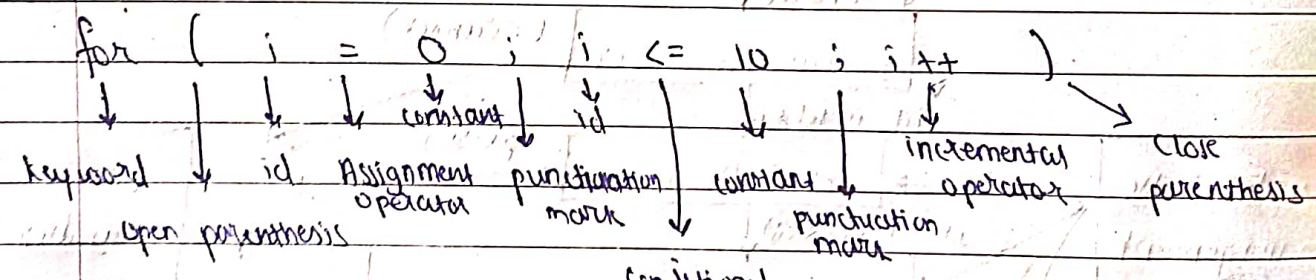
* Left to right Scanning
* Then Separation of continuous expression into single element which we will call it 'Token'.



* Step :- i) Scanning
ii) Tokenization
iii) Naming of each Token. (Identifier))

24/01/2023

for (i = 0; i <= 10; i++)



④ Syntax Analyzer :→

* Syntax Analyzer → check syntax.
* Semantic Analyzer → check logic

⑤ Intermediate code generation :→

Three addression code

$c = a + b$

ie. $id = id + (id * id)$

id

ie. $id = id + id$ ∵ $(id * id)$ replaced by id.