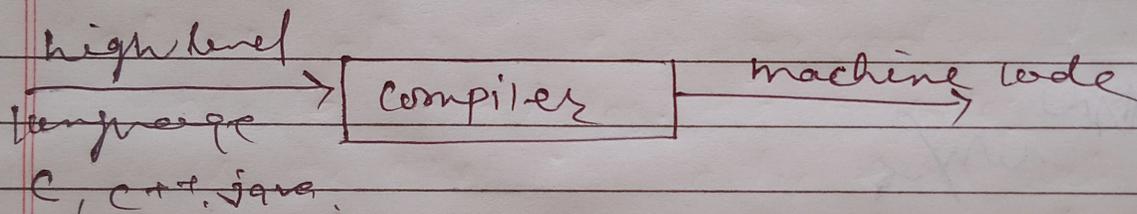


introduction to compilers.

A compiler is a system software which convert a code written in high level language to machine language which is understood by our computer. This process is called compilation, during compilation it report error present in code. If any error in source code it does not convert the code in machine language. It report only syntax error present in source code not for logical error.



The first high-level compiler is developed in 1954 by a team lead by John Backus at IBM for FORTRAN (Formula Translation).

Lexical Analysis: This is the first phase of compiler design. In this phase the compiler scan the source code from left to right. (So it is called scanner) character by character. It eliminate white space, comment present in source code and report any error present. It generate tokens which is pass to next phase of

Compiler -

for example:

```
int a, b = 10;
```

The above code pass to compiler, then lexical analysis scan the code from left to right and generate tokens as:

int → data type

a → variable

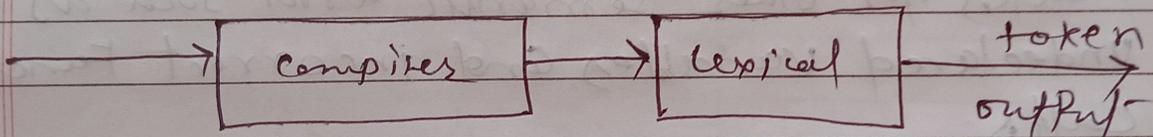
,

b → variable

= → Assignment operator

10 → constant

;



2) Syntax Analysis: It is second phase of compiler. It is called parser. It accept tokens generated by lexical analysis as input and create a parse tree. It check the code is syntactically correct or not whether the code follow the syntax of programming language or not. It report syntax error.

$x + y = a$ — ①

$a = x + y$ — ②

① Statement is syntax error.

② Statement is correct follow the syntax rule.

