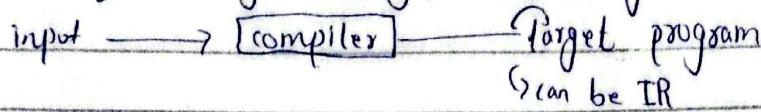


→ Compiler-construction: a compiler is a computer program that translates a program written in one language (source code) into an equivalent program (target lang) in another lang.

Input

in the form of high level lang, like, c,c++,java



means Intermediate code Representation

or Assembly language.

→ Compiler construction: The process of constructing a computer is called compiler construction.

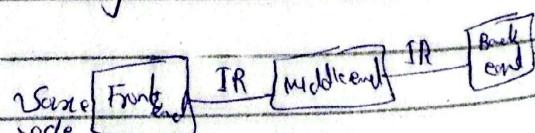
→ Why we construct compilers: without compilers, our machines (computers) are not able to understand the source code, written in high level lang.

→ Types of compilers:

1, Single pass	2, Multi Pass.
it compiles the whole program in only one-pass. ex: x86	it compiles the program source code multiple times. ex: java

Source code **Compiler** → Target code

Means it does not generate the PR



→ cousins of compilers. The phases of compiler construction is known as phases of compilers and that process is also known as language processing system.

→ 1st cousin:

Pre processor:

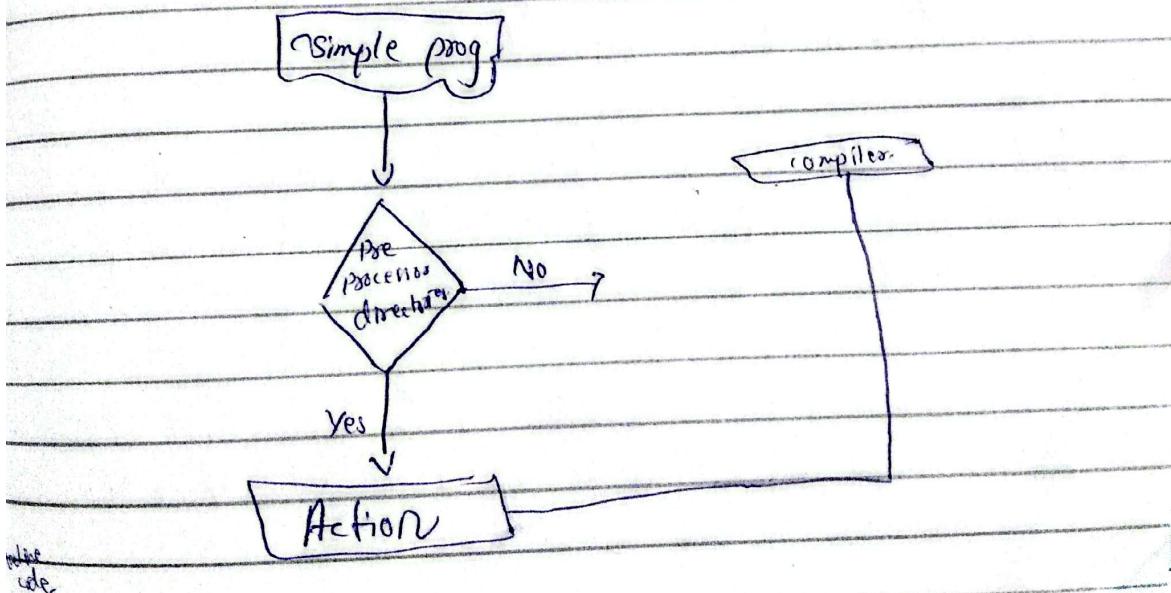
The pre-processor accepts source code as input and is responsible for:

→ It removes comments.

→ Process all preprocessor directives denoted by #.

means whatever # symbols will be included in our code, it will replace it with original coding.

→ diagram.



then that code comes as an input to the compiler.

② Compiler: it takes pure HLL as input and convert it into assemble codes

③ Assembler: it takes as a input assemble code and convert it into the relocated machine code.

④ Linking and loading:
it perform the four functions:

① Allocations: The relocatable machine code space will acquired and in memory that space will be assigned

② Relocations: All tasks will be loaded in main memory from secondary to main memory whose tasks priority is high.

③ Linker: it is a tool used to link all parts of program together for execution into a single executable file.

④ Loader: A Loader loads this executable file into the memory and starts its execution.

→ Phases of compiler.

There are six phases involve to construct the compiler.

→ source code. After removing directives the code comes in the first phase of compiler as an input to the first phase of compiler.

→ Lexical Analysis. It collects all the tokens and generate the output as stream of tokens and these stream of tokens goes to second phase.

2, Syntax Analysis phase (error reporting).: parse tree generate as a output and it will go to semantic Analysis Phase.

3, Semantic Analysis phase (error reporting) then it will convert into the syntax tree. 4phase.

4, Syntax tree. Then

5, IR (Intermediate code generation,

6, PR (Intermediate ~~code generation~~ representation)

7, code optimizer

8, code optimized

9, Target code (generation).