Pls kindly remember

1. Compiler and its types

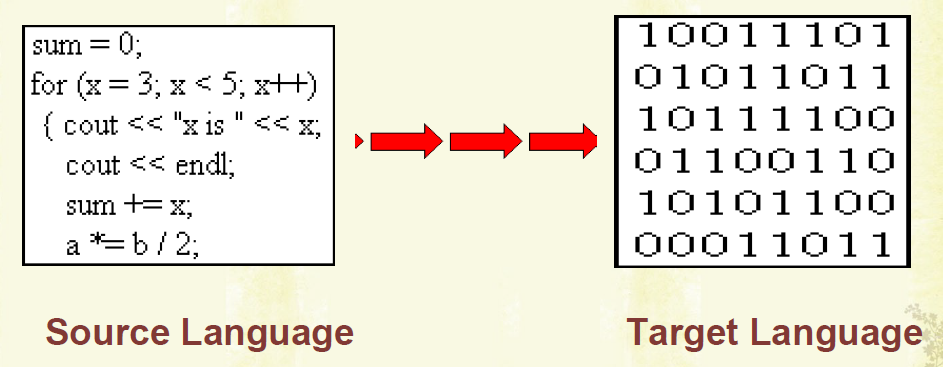
2. Language processing system (or) Cousin of compiler

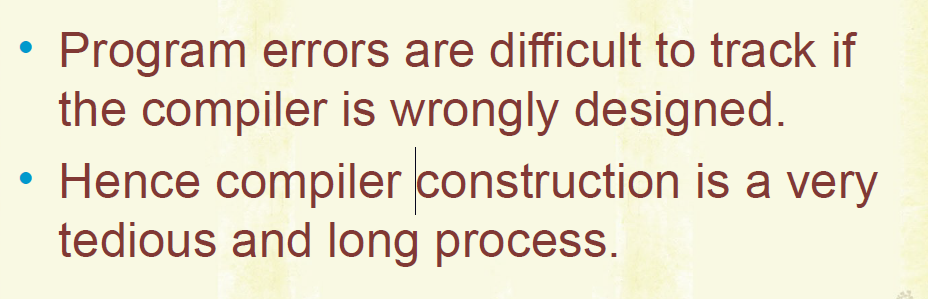
3.types of compiler

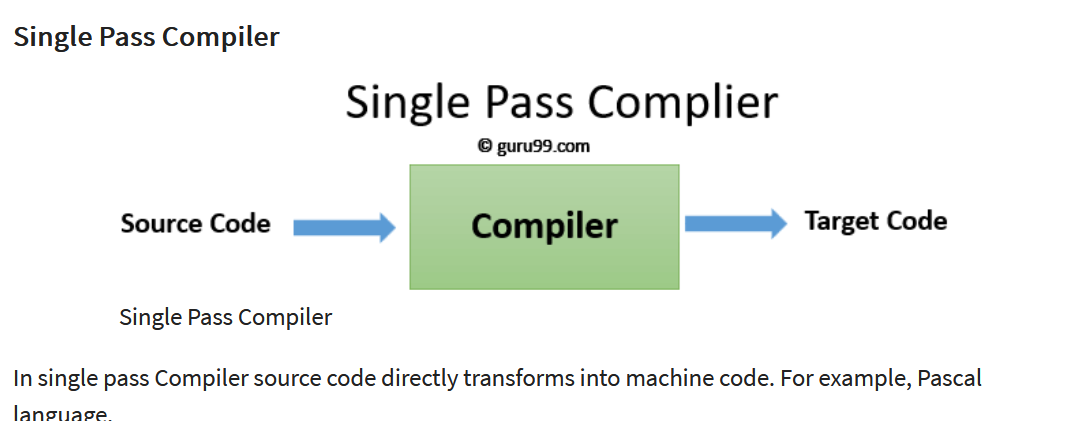
4.front end & back end

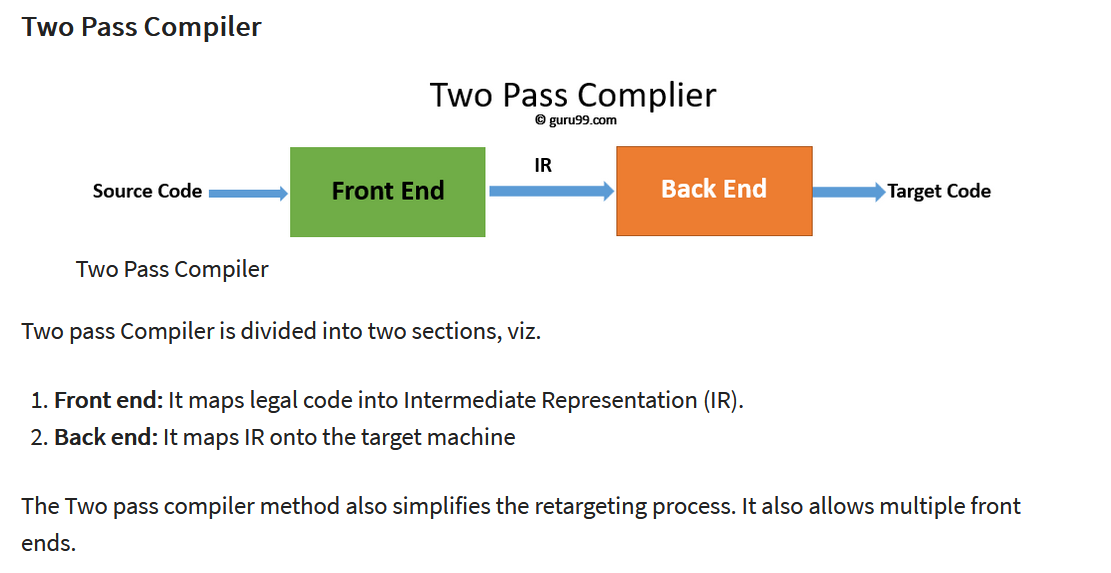
5.copiler vs interpreter.

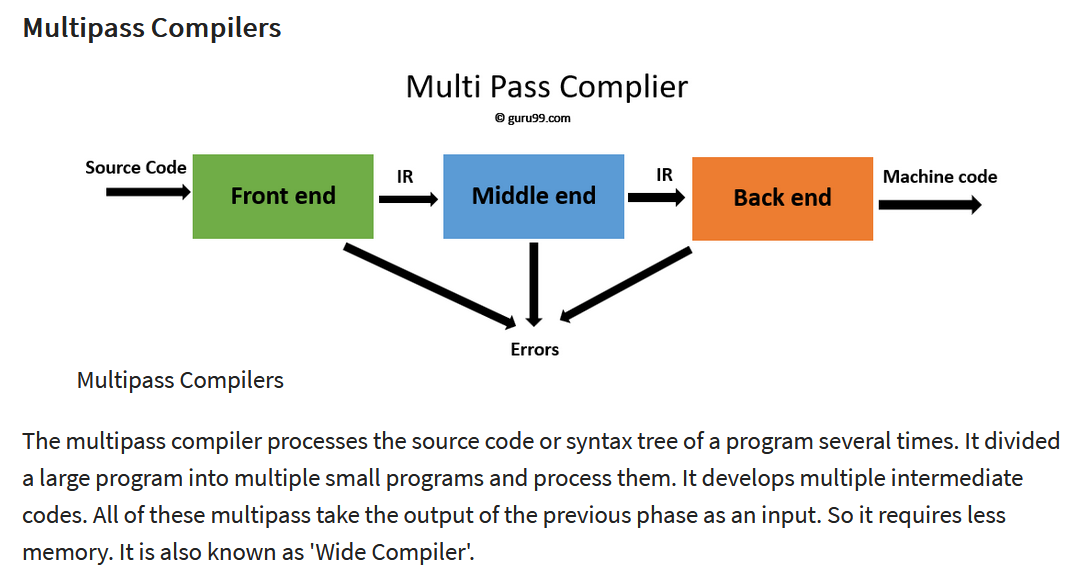
**1) Define compiler and its types**











**2) Explain language processing system/cousins of compiler**

**3) Explain different phases of compiler and what are the two parts of the compiler??**

[**http://www.brainkart.com/article/Cousins-of-Compiler\_8071/**](http://www.brainkart.com/article/Cousins-of-Compiler_8071/)

[**https://www.guru99.com/compiler-design-tutorial.html**](https://www.guru99.com/compiler-design-tutorial.html)

# **Phases of the compiler**

Lexical Analyser  
Convert high level language 🡪 stream of tokens  
🡪 Will remove the white spaces, comments

Syntax Analyser (Parser)  
stream of tokens 🡪 parse tree

Using the context free grammar, our high-level language must be matched.  
We have to check whether the given high-level language is matched with the context-free grammar. For checking the syntax analyser is going to construct a parser tree.

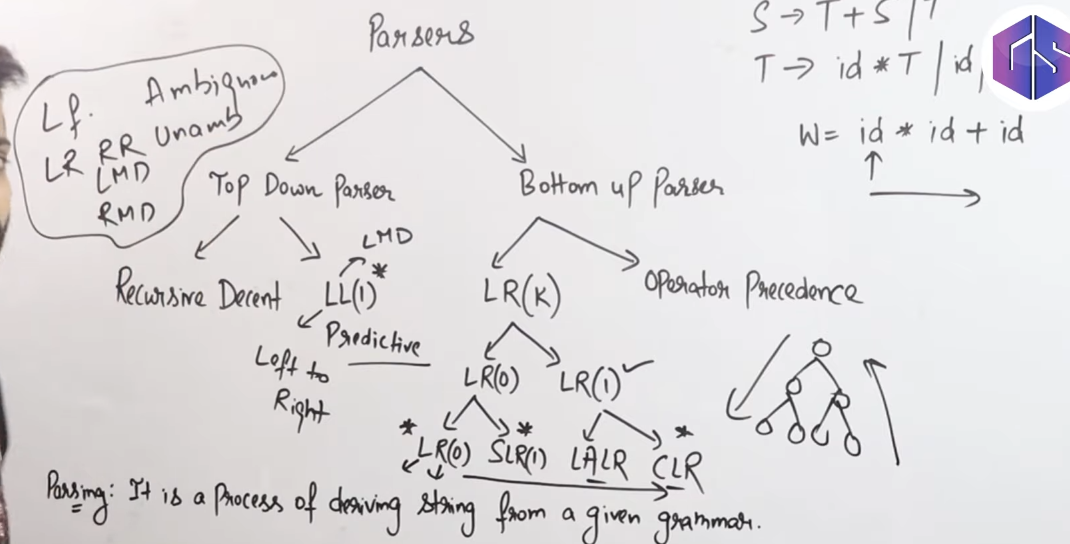
We have to check whether the yield of the parse tree and input to the syntax analyser are same.  
If same 🡪 No syntax error  
If not same 🡪 Syntax error

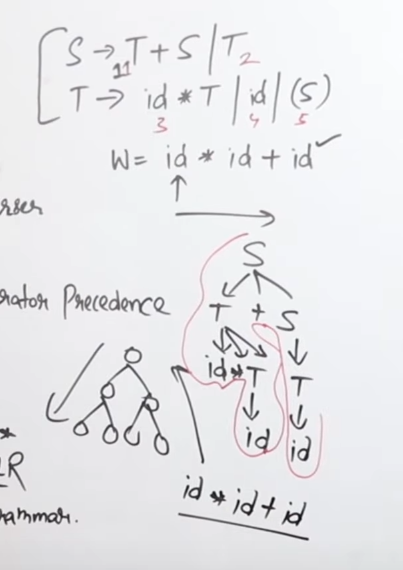
Intermediate Code Generator (ICG)  
Will reduce the entire program into 3 lines.  
Suppose, if we have a compiler for a specific platform which will run on PC, we can also design a compiler for the same platform which will run on mobile.  
Up-to Code Optimiser, the same thing for both PC based compiler and Mobile based compiler.  
Code-optimiser and target code generator must be changed.

# **Lexical Analyser**

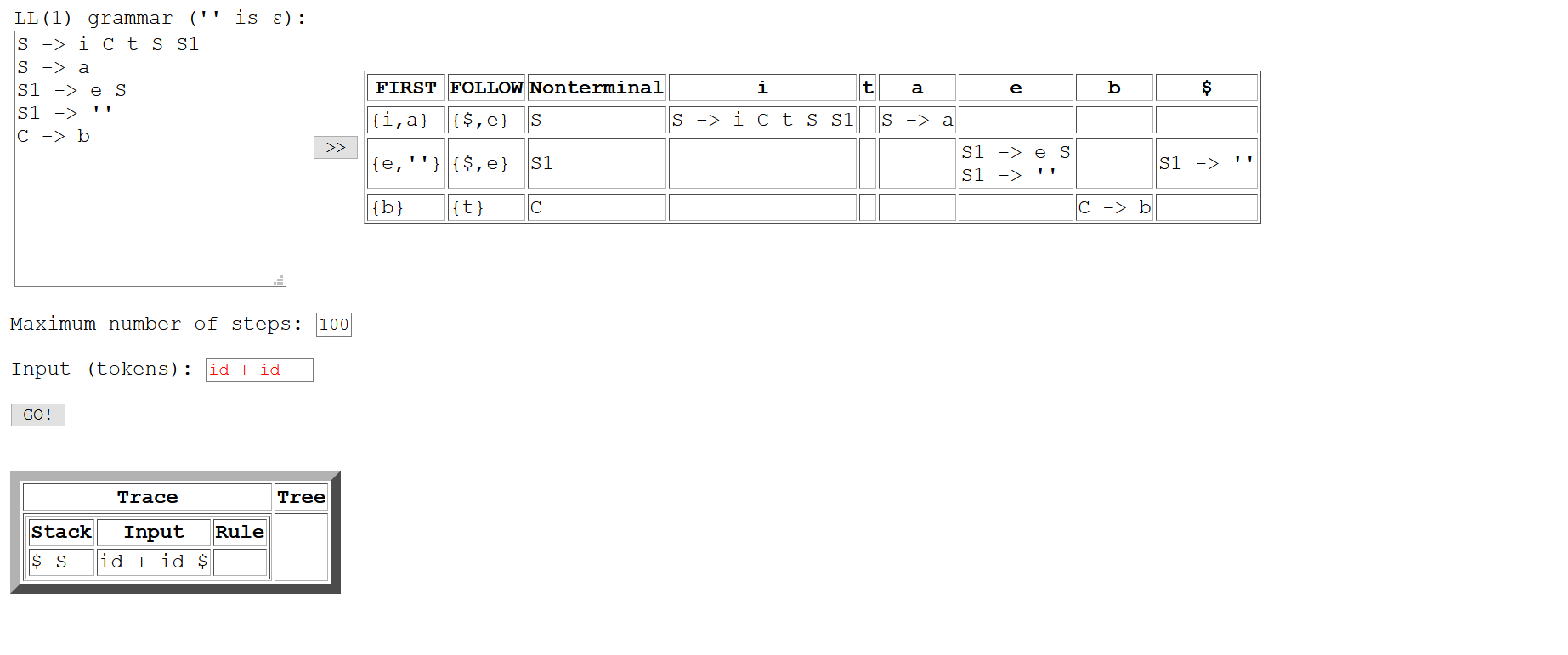
# **First()**

# **Follow()**





# **Short-cut to find whether a grammar is LL(1) or Not**



Each parser will have different-different parse table