What is compile?

A compiler is a computer program which helps you transform source code written in a high-level language into low-level machine language. It translates the code written in one programming language to some other language without changing the meaning of the code. The compiler also makes the end code efficient which is optimized for execution time and memory space.

There are 6 phases in a compiler. Each of this phase help in converting the high-level langue the machine code. The phases of a compiler are:

1. Lexical analysis
2. Syntax analysis
3. Semantic analysis
4. Intermediate code generator
5. Code optimizer
6. Code generator

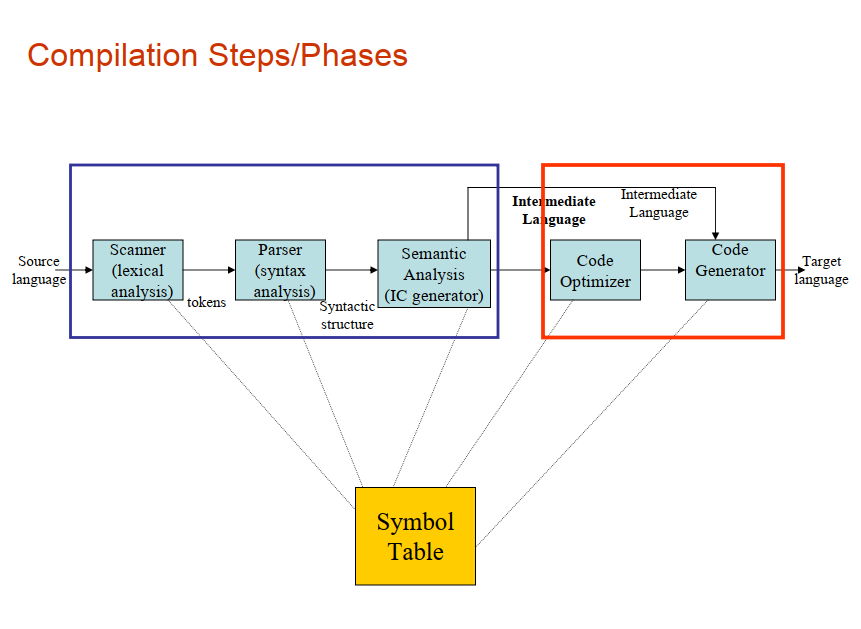
**Features of Compilers**

* Correctness
* Speed of compilation
* Preserve the correct the meaning of the code
* The speed of the target code
* Recognize legal and illegal program constructs
* Good error reporting/handling
* Code debugging help

**Types of Compiler**

Following are the different types of Compiler:

* Single Pass Compilers
* Two Pass Compilers
* Multipass Compilers



Lexical Analysis is the first phase of the compiler also known as a scanner. It converts the High level input program into a sequence of **Tokens**.

* The output is a sequence of tokens that is sent to the parser for syntax analysis

**What is a token?**  
A lexical token is a sequence of characters that can be treated as a unit in the grammar of the programming languages.

Different tokens or lexemes are:

* Keywords
* Identifiers
* Operators
* Constants

**Example of tokens:**

* Type token (id, number, real, . . .)
* Punctuation tokens (IF, void, return, . . .)
* Alphabetic tokens (keywords)

Keywords; Examples-for, while, if etc.

Identifier; Examples-Variable name, function name, etc.

Operators; Examples '+', '++', '-' etc.

Separators; Examples ',' ';' etc.

**Example of Non-Tokens:**

* Comments, preprocessor directive, macros, blanks, tabs, newline, etc.

