

**Lecture 01****1. What is compiler design?**

Compiler design is the process of developing a program or software that converts human-written code into machine code.

**2. What is a computer language translator?**

Computer translators transform human-readable code into machine-executable instructions.

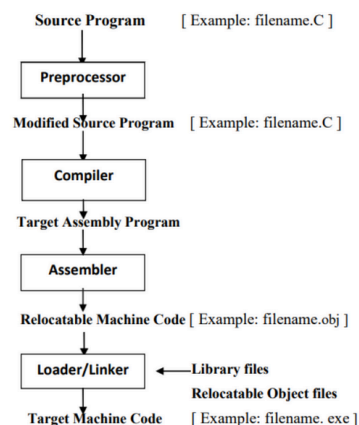
**3. Write the types of translator?**

There are three types of translator

- a. Assembler
- b. Compiler
- c. Interpreter

**4. Write the difference between compiler and interpreter?**

Compiler	Interpreter
Compiler converts high level language to low level language and vise-versa.	Interpreter converts high level language to low level language and vise-versa.
Compiler converts the whole code into machine code at a time.	Compiler converts one line into machine code at a time.
It takes less execution time in conversion compared to compiler.	It takes more time than a compiler.
Show error after whole code convert.	Show error after on line code convert.
<b>Examples:</b> C, C++, Java compilers, etc.	<b>Examples:</b> GAS, GNU assemblers.

**5. Draw the language processing system?**

## 6. What is a preprocessor?

**Pre-Processor:** Preprocessors prepare input files for other programs, often by expanding macros, including files, and conditionally compiling code.

## 7. Write difference between Linker and Loader?

Basic Comparison	Linker	Loader
Input	Object files generated by the compiler/assembler.	Executable file generated by the linker.
Function	Combines multiple object files into a single executable file.	Loads the executable file into memory.
Output	Executable file that can be run by the operating system.	Running program in memory.

## Lecture 02

### 1. What are compiler phases?

A compiler is a software program that converts the high-level source code written in a programming language into low-level machine code that can be executed by the computer hardware. The process of converting the source code into machine code involves several phases or stages, which are collectively known as the phases of a compiler.

### 2. Write the definition of lexical analysis with an example?

Example:  
x = y + 10

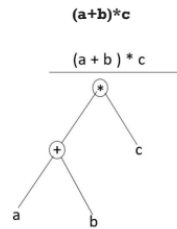
Tokens

x	identifier
=	Assignment operator
y	identifier
+	Addition operator
10	Number

The first phase of a compiler is lexical analysis, also known as scanning. This phase reads the source code and breaks it into a stream of tokens, which are the basic units of the programming language.

### 3. Write the definition of syntax analysis with an example?

The second phase of a compiler is syntax analysis, also known as parsing. This phase takes the stream of tokens generated by the lexical analysis phase and checks whether they conform to the grammar of the programming language.



**4. Write the phases of a compiler.**

- a. Lexical analysis
- b. Syntax analysis
- c. Semantic analysis
- d. Intermediate code generation
- e. Optimization
- f. Code generation

**5. Write the operation of the symbol data table?**

- Insertion of an item in the symbol table.
- Deletion of any item from the symbol table.
- Searching for the desired item from the symbol table.

**6. Write the error handler process.**

In the compiler design process error may occur in all the below-given phases:

- \* Lexical analyzer: Wrongly spelled tokens
- Syntax analyzer: Missing parenthesis
- Intermediate code generator: Mismatched operands for an operator
- Code Optimizer: When the statement is not reachable
- Code Generator: When the memory is full or proper registers are not allocated
- Symbol tables: Error of multiple declared identifiers

**7. Write the process of compilation.**

There are two phases of compilation. Analysis (Machine Independent)  
Synthesis (Machine Dependent)

**1. Analysis Phase**

- a) Lexical analysis
- b) syntax analysis
- c) Semantic analysis
- d) Intermediate Code Generation

**2. Synthesis Phase**

- a) Code Optimization
- b) Code Generation

**8. Write the definition of compiler pass and types?**

A Compiler pass refers to the traversal of a compiler through the entire program.

Compiler pass are two types:

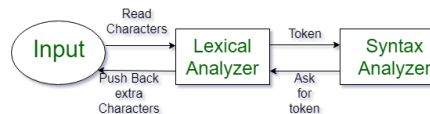
- Single Pass Compiler
- Two Pass Compiler or Multi Pass Compiler.

### Lecture 03

**1. Write the basic terminologies of lexical analysis with examples.**

- Token: It is basically a sequence of characters that are treated as a unit as it cannot be further broken down.
- Pattern: It specifies a set of rules that a scanner follows to create a token.
- Lexeme: It is a sequence of characters in the source code that are matched by given predefined language rules for every lexeme to be specified as a valid token.

**2. Draw the lexical analyser architecture?**



**3. Write the definition and example of the token.**

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

**Example of tokens:**

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

**4. Write the roles of lexical analyser.**

Lexical analyzer performs below given tasks:

- Helps to identify token into the symbol table
- Removes white spaces and comments from the source program
- Recognizing keywords and identifiers.
- Helps you to expands the macros if it is found in the source program
- Read input characters from the source program

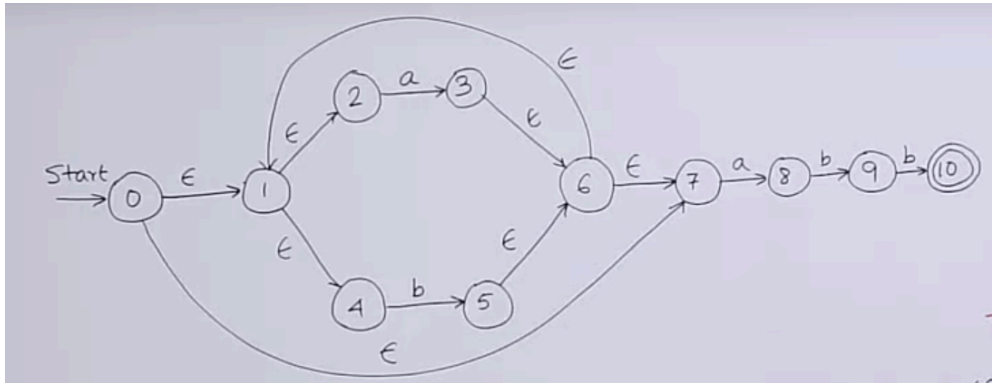
**5. Write the definition of lexical errors?**

A character sequence which is not possible to scan into any valid token is a lexical error.

### Lecture 05

1. Convert NFA to DFA of  $(a|b)^*abb$

Step 01:



Step 02:

$$\begin{aligned}
 A &= \epsilon\text{-closure}(\{0\}) = \{0, 1, 2, 4, 7\} \\
 B &= \epsilon\text{-closure}(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\} \\
 C &= \epsilon\text{-closure}(\{5\}) = \{1, 2, 4, 5, 6, 7\} \\
 D &= \epsilon\text{-closure}(\{5, 9\}) = \{1, 2, 4, 5, 6, 7, 9\} \\
 E &= \epsilon\text{-closure}(\{5, 10\}) = \{1, 2, 3, 5, 6, 7, 10\}
 \end{aligned}$$

Step 04:

