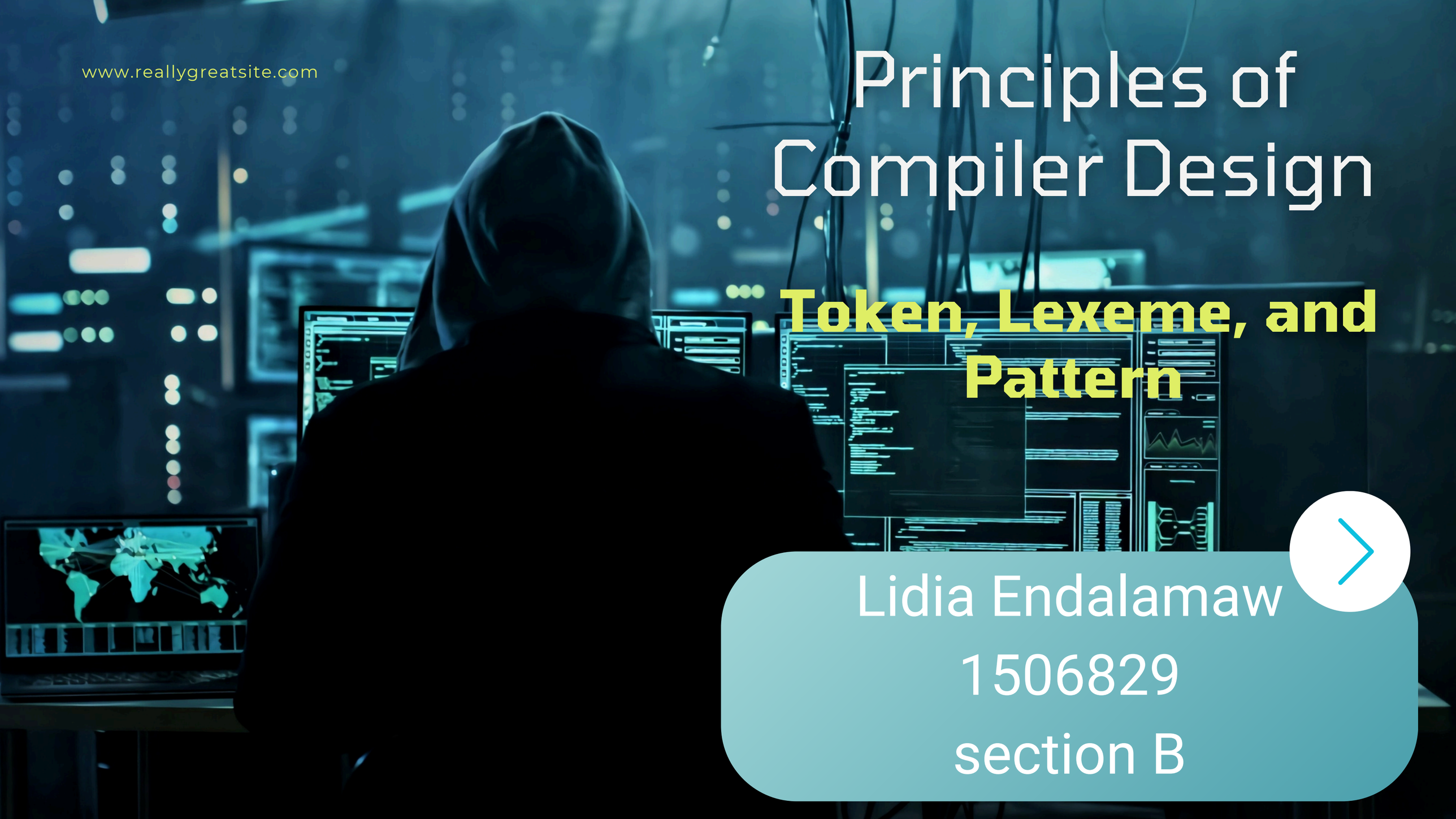

www.reallygreatsite.com

Principles of Compiler Design

Token, Lexeme, and Pattern

Lidia Endalamaw
1506829
section B

2. Introduction

In compiler design, the first phase of the compiler is Lexical Analysis. During this phase, the compiler reads the source code and breaks it into small meaningful units called tokens. To understand this process, we must know token, lexeme, and pattern.

These concepts help the compiler:

- Recognize keywords, identifiers, and operators
- Detect errors early
- Prepare input for the syntax analyzer

3. What is a Token?

A token is a category or type of meaningful unit in a program.
It represents what kind of element a part of the code is.

Examples of tokens

if → KEYWORD

x → IDENTIFIER

+ → OPERATOR

100 → NUMBER

if → KEYWORD

4. What is a Lexeme?

A lexeme is the actual sequence of characters in the source code that matches a token.
It is the real text written by the programmer.

Example Code:

- int count = 10;

Lexeme	Token Type
int	KEYWORD
count	IDENTFIER
=	ASSIGNMENT

5. What is a Pattern?

A pattern is a rule or expression that describes how a token looks.
Patterns are usually written using regular expressions.
Examples of Patterns:
IDENTIFIER → letter (letter | digit)*
NUMBER → digit+
KEYWORD → fixed words like if, while, int
The lexical analyzer uses patterns to recognize lexemes and assign tokens.

6. Relationship Between Token, Lexeme, and Pattern

TERM	MEANING
Token	Catagory or Type
Lexeme	Actual text in program
Pattern	Rule to match lexeme

- Example
  sum = total + 5;

LEXEME	TOKEN	PATTERN
Sum	IDENTIFIER	letter(letter
=	ASSIGN	=
5	Number	digit+

7. Role in Compiler Design

- Used in Lexical Analysis phase
- Helps convert source code into tokens
- Helps convert source code into tokens
- Improves error detection

Without tokens, lexemes, and patterns, the compiler cannot understand the program structure.

8. Real-World Application

Helps in building compilers and translators

01 Used in programming languages like C, Java, Python

02 Used in IDEs for syntax highlighting

03 Used in code analyzers and interpreters

9. Conclusion

A token represents the type of language element

A lexeme is the actual text in the source code

A pattern defines the rule for forming tokens

Together, they form the foundation of lexical analysis in compiler design.

10. References

1.Alfred V. Aho et al., Compilers: Principles, Techniques, and Tools.

3.Kenneth C. Louden, Compiler Construction

3.GeeksforGeeks – Compiler Design (Lexical Analysis)

Thank You ❤️