The General Problem of Describing Syntax

A Language, whether natural-like English, or artificial
like C, Java, is a set of strings of characters from
some alphabet. The strings of a language are called
Sentences or Statements. So Syntax Rules of a
language specify which strings of characters from the
language’s alphabet are in the language.

Formal descriptions of the Syntax of programming languages, for
Simplicity’s sake, often do not include descriptions of the
lowest-level syntactic units. These small units are called
Lexemes.

Lexemes include its numeric literals, operators, and special
words, among others. We can think of programs as
Strings of lexemes, rather than of characters.

Lexemes are partitioned into groups—for example, the names
of variables, methods, classes, and so forth, in a programming
language, form a group called identifiers.

Each lexeme group is represented by a name, or token. So, a
token of a language is a category of its lexemes.

Computer Science Lectures By ER. Deepak Garg
Consider the following Java statement:

```java
index = 2 * Count + 17;
```

<table>
<thead>
<tr>
<th>Lexemes</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>identifiers</td>
</tr>
<tr>
<td>=</td>
<td>equal_sign</td>
</tr>
<tr>
<td>2</td>
<td>int_literal</td>
</tr>
<tr>
<td>*</td>
<td>mult_op</td>
</tr>
<tr>
<td>Count</td>
<td>identifies</td>
</tr>
<tr>
<td>+</td>
<td>plus_op</td>
</tr>
<tr>
<td>17</td>
<td>int_literal</td>
</tr>
<tr>
<td>;</td>
<td>Semicolon</td>
</tr>
</tbody>
</table>

2 Distinct ways of defining a language:

A) **Language Recognizers**:

- A recognition device reads input strings of the language and decides whether the input strings belong to the language.

B) **Language Generators**:

- It generates sentences of a language.
- People prefer certain forms of Generators over Recognizers because they can more easily read and comprehend.

Computer Science Lectures By ER. Deepak Garg
Understand them.

By contrast, the syntax-checking portion of a compiler (a language recognizer) is not as useful as language description for a programmer because it can be used only in trial-and-error mode.

To determine correct syntax of a particular statement using a compiler, the programmer can only submit speculated version and note whether the compiler accepts it.

On the other hand, it is often possible to determine whether the syntax of a particular statement is correctly comparing it with the structure of the generator.