

COMP205
Comparative Programming
Languages

Part 1: Introduction to programming languages

Lecture 2: Structure of programs and programming languages as communication media

PROGRAM STRUCTURE, AND PROGRAMMING LANGUAGES AS COMMUNICATION MEDIA

1. The structure of programs

- Lexical structure
- Syntactic structure
- Context and semantics

2. Programming languages as communication media

THE STRUCTURE OF COMPUTER PROGRAMS

We can analyse a computer program on 4 levels:

1. Lexical
2. Syntactic
3. Contextual
4. Semantic

LEXICAL STRUCTURE

- The lexical level is the lowest level.
- At this level computer programs are viewed as simple sequences of lexical items called *tokens*.
- We can classify tokens as being either:

- | | |
|----------------|---------------|
| 1. Identifiers | 4. Separators |
| 2. Keywords | 5. Literals |
| 3. Operators | 6. Comments |

Identifiers: Names chosen to represent data items, functions and procedures, etc.

Considerations:

- Case sensitivity
- Number of characters

Keywords: Names chosen by the language designer to represent facets of particular language constructs which cannot be used as identifiers (sometimes referred to as *reserved* words).

Operators: Special “keywords” used to identify operations to be performed on *operands*, e.g. maths operators.

Separators: Punctuation marks used to group together sequences of tokens that have a “unit” meaning.

When outputting text it is often desirable to include punctuation, where these are also used (within the language) as separators we must precede the punctuation character with what is called an *escape character* (usually a backslash ‘\’).

Literals:

Denote direct values, can be:

- ❑ Numeric, e.g. 1, -123, 3.14, 6.02e23.
- ❑ Character, e.g. 'a'.
- ❑ String, e.g. "Some text".

Comments:

A good program is one that is understandable. We can increase understandability by including meaningful comments into our code.

Comments are omitted during processing.

The start of a comment is typically indicated by a keyword (e.g. comment) or a separator, and may also be ended by a separator.

LAYOUT

- Generally speaking layout (indentation etc.) is unimportant in the context of programming languages.
- **White space** (spaces, tabs and new lines) are usually ignored.
- A good layout does however enhance readability, and consequently understandability.
- A good layout can also reduce the risk of programming errors.

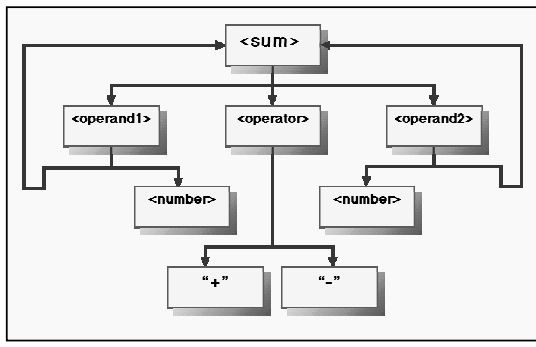
SYNTACTIC STRUCTURE

- The syntactic level describes the way that program statements are constructed from tokens.
- This is always very precisely defined in terms of a **context free grammar**.
- The best known examples are BNF (Backus Naur Form) or EBNF (Extended Backus Naur Form).
- Syntax may also be described using a **syntax tree**.

EBNF EXAMPLE

```
<sum> ::= <operand>
        <operator>
        <operand>
<operand> ::= <number> |
              ( <sum> )
<operator> ::= + | -
```

SYNTAX TREE



CONTEXT

- The contextual level of analysis is concerned with the "context" in which program statements occur.
- Program statements usually contain identifiers whose value is dictated by earlier statements (especially in the case of the imperative or OO paradigms).
- Consequently the meaning of a statement is dependent on "what has gone before", i.e. its context.
- Context also determines whether a statement is "legal" or not (**context conditions** – a data item must "exist" before it can be used).

SEMANTICS

- The semantic level refers to the final overall meaning of a program.

PROGRAMMING LANGUAGES AS COMMUNICATION MEDIA

- Like any other language programming languages are essential communications media:
 - human to machine,
 - machine to machine, and
 - machine to human.
- Communication is about readability!
- Unlike humans, machines are not fault tolerant!
- Hence programming is often viewed as a “complex” task.

SUMMARY

- Structure of languages:
 1. Lexical
 2. Syntactic
 3. Contextual
 4. Semantic
- Programming languages as communications media.