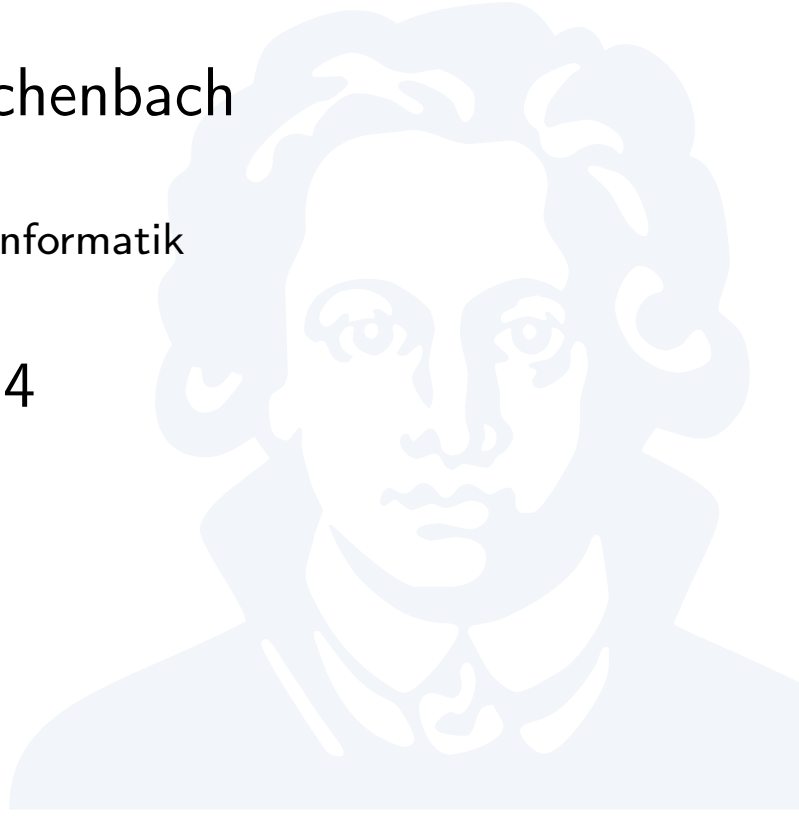
Foundations of Programming Languages

Statements and Assignments

Prof. Dr. Christoph Reichenbach

Fachbereich 12 / Institut für Informatik

22. Oktober 2014



Statements

- Commands
- Classes of statements:
  - Assignments
  - Blocks
  - Conditional statements
  - Loop statements
  - Subprogram calls
  - promoted declarations
  - promoted expressions
- Functional programming:
  - No statements, or
  - statements = expressions

Blocks

Pascal

```
begin
    statement1;
    statement2;
end
```

C family

```
{
    statement1;
    statement2;
}
```

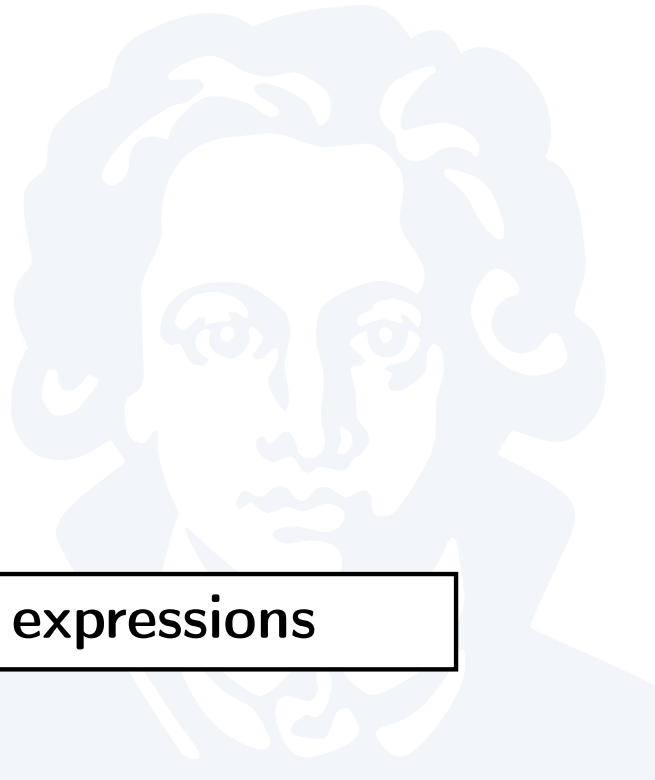
- Alternative approach (Python, Haskell):
  use *indentation/whitespace* differences to mark blocks

Combine multiple statements into a single statement

Assignments

`x := expression`

- Evaluate: *expression* $\Downarrow v$
- Put v into `x`'s storage location
- Multiple assignment (e.g., Python):
  `(x, y) = (y, x)`
- Common assignment operators:
  - `:=` in Pascal, SML
  - `=` in the C family
  - `<-` in OCaml



In C-like languages, assignments are expressions

Compound Assignment

Common feature in C family languages:

- `x += 3`
  equivalent to
  `x = x + 3`
- Analogously for -=, *=, <<=, ...

Unary Assignment

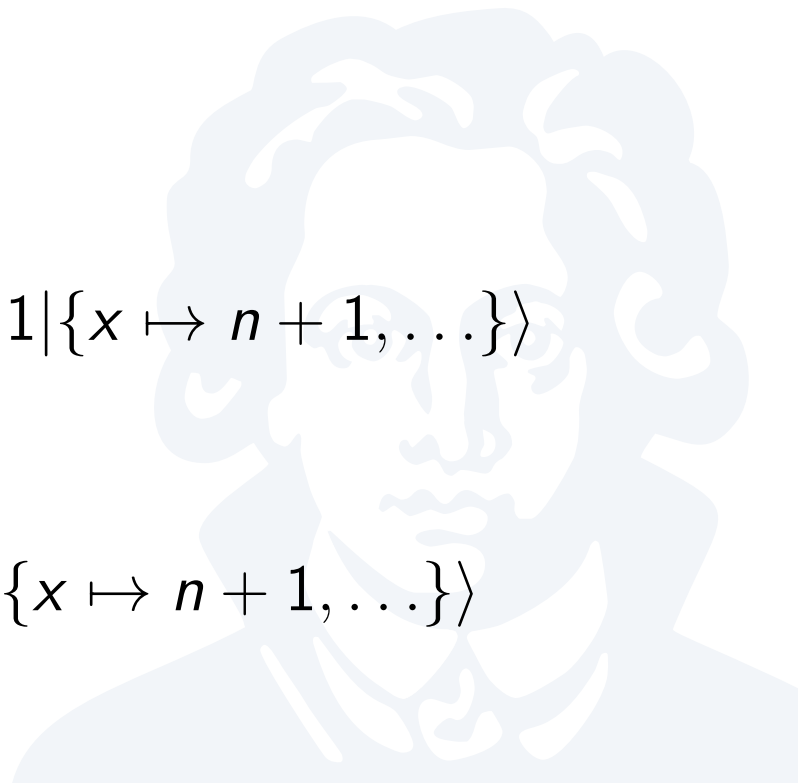
Another common feature in C family languages:

- `++x`
  equivalent to:
  `x = x + 1`
- `--x`
  equivalent to
  `x = x - 1`
- Also: `x++`, `x--`
- `++x` returns new `x`:

$$\langle \texttt{++x} | \{x \mapsto n, \ldots\} \rangle \longrightarrow^{\star} \langle n+1 | \{x \mapsto n+1, \ldots\} \rangle$$

- `x++` returns old `x`:

$$\langle \texttt{x++} | \{x \mapsto n, \ldots\} \rangle \longrightarrow^{\star} \langle n | \{x \mapsto n+1, \ldots\} \rangle$$



Summary

- *Statements*: orders to the computer (sequential)
- *Blocks*: combine multiple statements into one
- *Assignments* update storage location of a variable
  - *Multiple assignments*: assign to multiple variables at once
  - *Compound assignments*: combine old variable contents with value arithmetically
  - *Unary assignment*: increment/decrement variable