

Solutions

Each answer shows the steps, then the **result**.

2.1

- a **compiler** translates the **whole** program to machine code first (producing an executable)
- an **interpreter** translates and **runs each statement** as it goes (no executable)

2.2

- **lexical analysis** → **syntax analysis (parsing)** → **semantic analysis**

2.3

- **tokens** (keywords, identifiers, operators, literals)

2.4

- 3 4 +

3.1

- any four: it reads a statement, does **lexical** then **syntax** analysis
- checks it / its types
- **executes** the statement's action immediately
- reports any error at once and usually stops
- repeats for each statement — no executable is produced — max 4

3.2

- $\langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

3.3

- 3 4 + for the bracket
- 2 * last: 3 4 + 2 *

3.4

- push 5, push 2; - → pop 2 and 5, $5 - 2 = 3$, push 3
- push 3 (stack 3, 3)
- * → $3 \times 3 = 9$, result **9**

3.5

- to make the object (machine) code run **faster** and/or use **less memory**, without changing the result it produces