08 Programming

SrWL Learning · Complete Classroom Teaching Version · 2026–2028 target

Write and maintain working programs

55 editable slides / Explanation · Visual · Practice · Teacher Check

By the end of this chapter

- Variables and data types.
- Selection and iteration.
- Arrays and subroutines.
- Files and debugging.

Learning goals are organised into bite-sized lessons.

Our learning route

10 mini-lessons · each includes one guided check

- 01 Variables, constants and data types: A
- 02 Variables, constants and data types: B
- 03 Input, output and control: A
- 04 Input, output and control: B
- 05 Strings and operators: A
- 06 Strings and operators: B
- 07 Arrays, procedures and functions: A
- 08 Arrays, procedures and functions: B
- 09 File handling and maintenance: A
- 10 File handling and maintenance: B

01 Variables, constants and data types · Foundations

One clear idea to explain before moving to practice

Variables store values that may change.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Variables, constants and data types — key points

- Variables store values that may change.
- Constants store values that should remain fixed.

Worked context: Example: Total ← Price * Quantity.

Variables, constants and data types — connect the ideas

Follow the concept from left to right

01

Variables store values that may change.

→

02

Constants store values that should remain fixed.

Apply the principles to this example: Example: Total ← Price * Quantity.

Try it · Variables, constants and data types

Think (30 s) · Pair discussion (1 min) · Share

Explain the meaning of this idea and give a relevant example:

Constants store values that should remain fixed.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Variables, constants and data types

Model response · Correct misconceptions

Constants store values that should remain fixed.

EXAM TIP / EXTENSION

Use this worked context to add detail: Example: Total ← Price * Quantity.

02 Variables, constants and data types · Applications

One clear idea to explain before moving to practice

Common types include INTEGER, REAL, STRING, CHAR and BOOLEAN.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Variables, constants and data types — key points

- Common types include INTEGER, REAL, STRING, CHAR and BOOLEAN.
- Expressions combine values, variables and operators.

Worked context: Example: Total ← Price * Quantity.

Variables, constants and data types — connect the ideas

Follow the concept from left to right

01

Common types include INTEGER, REAL, STRING, CHAR and BOOLEAN.

→

02

Expressions combine values, variables and operators.

Apply the principles to this example: Example: Total ← Price * Quantity.

Try it · Variables, constants and data types

Think (30 s) · Pair discussion (1 min) · Share

Describe when this principle applies:

Expressions combine values, variables and operators.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Variables, constants and data types

Model response · Correct misconceptions

Expressions combine values, variables and operators.

EXAM TIP / EXTENSION

Use this worked context to add detail: Example: Total ← Price * Quantity.

03 Input, output and control · Foundations

One clear idea to explain before moving to practice

INPUT receives values; OUTPUT displays information.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Input, output and control — key points

- INPUT receives values; OUTPUT displays information.
- Sequence executes operations in order.

Worked context: Write a program that counts how many values exceed 50.

Input, output and control — connect the ideas

Follow the concept from left to right

01

INPUT receives values; OUTPUT displays information.

→

02

Sequence executes operations in order.

Apply the principles to this example: Write a program that counts how many values exceed 50.

Try it · Input, output and control

Think (30 s) · Pair discussion (1 min) · Share

Identify the key technical terms and describe their roles: Sequence executes operations in order.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Input, output and control

Model response · Correct misconceptions

Sequence executes operations in order.

EXAM TIP / EXTENSION

Use this worked context to add detail: Write a program that counts how many values exceed 50.

04 Input, output and control · Applications

One clear idea to explain before moving to practice

IF / ELSE selects between paths.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Input, output and control — key points

- IF / ELSE selects between paths.
- FOR, WHILE and REPEAT control iteration.

Worked context: Write a program that counts how many values exceed 50.

Input, output and control — connect the ideas

Follow the concept from left to right

01

IF / ELSE selects between paths.

→

02

FOR, WHILE and REPEAT control iteration.

Apply the principles to this example: Write a program that counts how many values exceed 50.

Try it · Input, output and control

Think (30 s) · Pair discussion (1 min) · Share

Compare this concept with a related alternative:
FOR, WHILE and REPEAT control iteration.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Input, output and control

Model response · Correct misconceptions

FOR, WHILE and REPEAT control iteration.

EXAM TIP / EXTENSION

Use this worked context to add detail: Write a program that counts how many values exceed 50.

05 Strings and operators · Foundations

One clear idea to explain before moving to practice

Arithmetic operators compute numeric results.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Strings and operators — key points

- Arithmetic operators compute numeric results.
- Relational operators compare values.

Worked context: Example condition: Age >= 12 AND Age <= 16.

Strings and operators — connect the ideas

Follow the concept from left to right

01

Arithmetic operators compute numeric results.

→

02

Relational operators compare values.

Apply the principles to this example: Example condition: Age >= 12 AND Age <= 16.

Try it · Strings and operators

Think (30 s) · Pair discussion (1 min) · Share

Apply this idea to a new computer science scenario: Relational operators compare values.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Strings and operators

Model response · Correct misconceptions

Relational operators compare values.

EXAM TIP / EXTENSION

Use this worked context to add detail: Example condition: Age >= 12 AND Age <= 16.

06 Strings and operators · Applications

One clear idea to explain before moving to practice

AND, OR and NOT combine conditions.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Strings and operators — key points

- AND, OR and NOT combine conditions.
- String operations process characters and text.

Worked context: Example condition: Age >= 12 AND Age <= 16.

Strings and operators — connect the ideas

Follow the concept from left to right

01

AND, OR and NOT combine conditions.

→

02

String operations process characters and text.

Apply the principles to this example: Example condition: Age >= 12 AND Age <= 16.

Try it · Strings and operators

Think (30 s) · Pair discussion (1 min) · Share

Explain the meaning of this idea and give a relevant example:

String operations process characters and text.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Strings and operators

Model response · Correct misconceptions

String operations process characters and text.

EXAM TIP / EXTENSION

Use this worked context to add detail: Example condition: Age >= 12 AND Age <= 16.

07 Arrays, procedures and functions · Foundations

One clear idea to explain before moving to practice

A one-dimensional array stores an indexed list.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Arrays, procedures and functions — key points

- A one-dimensional array stores an indexed list.
- A two-dimensional array stores a grid.

Worked context: Students[1:30] stores 30 indexed entries.

Arrays, procedures and functions — connect the ideas

Follow the concept from left to right

01

A one-dimensional array stores an indexed list.

→

02

A two-dimensional array stores a grid.

Apply the principles to this example: Students[1:30] stores 30 indexed entries.

Try it · Arrays, procedures and functions

Think (30 s) · Pair discussion (1 min) · Share

Describe when this principle applies:

A two-dimensional array stores a grid.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Arrays, procedures and functions

Model response · Correct misconceptions

A two-dimensional array stores a grid.

EXAM TIP / EXTENSION

Use this worked context to add detail: Students[1:30] stores 30 indexed entries.

08 Arrays, procedures and functions · Applications

One clear idea to explain before moving to practice

Procedures perform a task; functions return a value.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Arrays, procedures and functions — key points

- Procedures perform a task; functions return a value.
- Parameters pass values into subroutines.

Worked context: Students[1:30] stores 30 indexed entries.

Arrays, procedures and functions — connect the ideas

Follow the concept from left to right

01

Procedures perform a task; functions return a value.

→

02

Parameters pass values into subroutines.

Apply the principles to this example: Students[1:30] stores 30 indexed entries.

Try it · Arrays, procedures and functions

Think (30 s) · Pair discussion (1 min) · Share

Identify the key technical terms and describe their roles: Parameters pass values into subroutines.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · Arrays, procedures and functions

Model response · Correct misconceptions

Parameters pass values into subroutines.

EXAM TIP / EXTENSION

Use this worked context to add detail: Students[1:30] stores 30 indexed entries.

09 File handling and maintenance · Foundations

One clear idea to explain before moving to practice

Files can store data beyond a program run.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

File handling and maintenance — key points

- Files can store data beyond a program run.
- Open the file before reading or writing.

Worked context: Read, process and output data from a text file.

File handling and maintenance — connect the ideas

Follow the concept from left to right

01

Files can store data beyond a program run.

→

02

Open the file before reading or writing.

Apply the principles to this example: Read, process and output data from a text file.

Try it · File handling and maintenance

Think (30 s) · Pair discussion (1 min) · Share

Compare this concept with a related alternative:

Open the file before reading or writing.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · File handling and maintenance

Model response · Correct misconceptions

Open the file before reading or writing.

EXAM TIP / EXTENSION

Use this worked context to add detail: Read, process and output data from a text file.

10 File handling and maintenance · Applications

One clear idea to explain before moving to practice

Close files after operations are complete.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

File handling and maintenance — key points

- Close files after operations are complete.
- Meaningful names, comments and testing aid maintenance.

Worked context: Read, process and output data from a text file.

File handling and maintenance — connect the ideas

Follow the concept from left to right

01

Close files after operations are complete.

→

02

Meaningful names, comments and testing aid maintenance.

Apply the principles to this example: Read, process and output data from a text file.

Try it · File handling and maintenance

Think (30 s) · Pair discussion (1 min) · Share

Apply this idea to a new computer science scenario: Meaningful names, comments and testing aid maintenance.

YOUR TASK

Use accurate technical vocabulary. Add a second point or a specific example where possible.

Teacher check · File handling and maintenance

Model response · Correct misconceptions

Meaningful names, comments and testing aid maintenance.

EXAM TIP / EXTENSION

Use this worked context to add detail: Read, process and output data from a text file.

Five essential ideas

Recall these without looking at the teaching slides

01 Variables, constants and data types

02 Input, output and control

03 Strings and operators

04 Arrays, procedures and functions

05 File handling and maintenance

End-of-chapter challenge

Answer in your notebook, then compare with a partner

How is a 2D array different from a 1D array?

TEACHER MODEL ANSWER

A 1D array uses one index; a 2D array uses two indices (row and column).