03 Hardware

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

How computer systems process and store data

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

By the end of this chapter

- CPU and registers.
- Fetch–decode–execute.
- Memory and storage.
- Input and output.

Learning goals are organised into bite-sized lessons.

Our learning route

10 mini-lessons · each includes one guided check

- 01 CPU architecture: A
- 02 CPU architecture: B
- 03 Registers to remember: A
- 04 Registers to remember: B
- 05 Fetch–decode–execute cycle: A
- 06 Fetch–decode–execute cycle: B
- 07 Memory, performance and storage: A
- 08 Memory, performance and storage: B
- 09 Input / output devices: A
- 10 Input / output devices: B

01 CPU architecture · Foundations

One clear idea to explain before moving to practice

The CPU executes instructions and coordinates processing.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

CPU architecture — key points

- The CPU executes instructions and coordinates processing.
- The control unit directs operations and timing.

Worked context: CPU = CU + ALU + registers (simplified model).

CPU architecture — connect the ideas

Follow the concept from left to right

01

The CPU executes instructions and coordinates processing.

→

02

The control unit directs operations and timing.

Apply the principles to this example: CPU = CU + ALU + registers (simplified model).

Try it · CPU architecture

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

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

The control unit directs operations and timing.

YOUR TASK

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

Teacher check · CPU architecture

Model response · Correct misconceptions

The control unit directs operations and timing.

EXAM TIP / EXTENSION

Use this worked context to add detail: CPU = CU + ALU + registers (simplified model).

02 CPU architecture · Applications

One clear idea to explain before moving to practice

The ALU carries out arithmetic and logic.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

CPU architecture — key points

- The ALU carries out arithmetic and logic.
- Registers store small amounts of data very quickly.

Worked context: CPU = CU + ALU + registers (simplified model).

CPU architecture — connect the ideas

Follow the concept from left to right

01

The ALU carries out arithmetic and logic.

→

02

Registers store small amounts of data very quickly.

Apply the principles to this example: CPU = CU + ALU + registers (simplified model).

Try it · CPU architecture

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

Describe when this principle applies:

Registers store small amounts of data very quickly.

YOUR TASK

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

Teacher check · CPU architecture

Model response · Correct misconceptions

Registers store small amounts of data very quickly.

EXAM TIP / EXTENSION

Use this worked context to add detail: CPU = CU + ALU + registers (simplified model).

03 Registers to remember · Foundations

One clear idea to explain before moving to practice

PC: address of the next instruction.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Registers to remember — key points

- PC: address of the next instruction.
- MAR: memory location currently being accessed.

Worked context: Different registers have different roles.

Registers to remember — connect the ideas

Follow the concept from left to right

01

PC: address of the next instruction.

→

02

MAR: memory location currently being accessed.

Apply the principles to this example: Different registers have different roles.

Try it · Registers to remember

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

Identify the key technical terms and describe their roles:
MAR: memory location currently being accessed.

YOUR TASK

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

Teacher check · Registers to remember

Model response · Correct misconceptions

MAR: memory location currently being accessed.

EXAM TIP / EXTENSION

Use this worked context to add detail: Different registers have different roles.

04 Registers to remember · Applications

One clear idea to explain before moving to practice

MDR: data being read from or written to memory.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Registers to remember — key points

- MDR: data being read from or written to memory.
- CIR: current instruction; ACC: intermediate calculation.

Worked context: Different registers have different roles.

Registers to remember — connect the ideas

Follow the concept from left to right

01

MDR: data being read from or written to memory.

→

02

CIR: current instruction; ACC: intermediate calculation.

Apply the principles to this example: Different registers have different roles.

Try it · Registers to remember

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

Compare this concept with a related alternative:
CIR: current instruction; ACC: intermediate calculation.

YOUR TASK

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

Teacher check · Registers to remember

Model response · Correct misconceptions

CIR: current instruction; ACC: intermediate calculation.

EXAM TIP / EXTENSION

Use this worked context to add detail: Different registers have different roles.

05 Fetch–decode–execute cycle · Foundations

One clear idea to explain before moving to practice

Fetch an instruction from main memory.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Fetch–decode–execute cycle — key points

- Fetch an instruction from main memory.
- Decode the instruction in the control unit.

Worked context: The cycle repeats while programs run.

Fetch–decode–execute cycle — connect the ideas

Follow the concept from left to right

01

Fetch an instruction from main memory.

→

02

Decode the instruction in the control unit.

Apply the principles to this example: The cycle repeats while programs run.

Try it · Fetch–decode–execute cycle

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

Apply this idea to a new computer science scenario: Decode the instruction in the control unit.

YOUR TASK

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

Teacher check · Fetch–decode–execute cycle

Model response · Correct misconceptions

Decode the instruction in the control unit.

EXAM TIP / EXTENSION

Use this worked context to add detail: The cycle repeats while programs run.

06 Fetch–decode–execute cycle · Applications

One clear idea to explain before moving to practice

Execute the specified operation.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Fetch–decode–execute cycle — key points

- Execute the specified operation.
- Update registers and continue with the next instruction.

Worked context: The cycle repeats while programs run.

Fetch–decode–execute cycle — connect the ideas

Follow the concept from left to right

01

Execute the specified operation.

→

02

Update registers and continue with the next instruction.

Apply the principles to this example: The cycle repeats while programs run.

Try it · Fetch–decode–execute cycle

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

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

Update registers and continue with the next instruction.

YOUR TASK

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

Teacher check · Fetch–decode–execute cycle

Model response · Correct misconceptions

Update registers and continue with the next instruction.

EXAM TIP / EXTENSION

Use this worked context to add detail: The cycle repeats while programs run.

07 Memory, performance and storage · Foundations

One clear idea to explain before moving to practice

RAM is volatile working memory.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Memory, performance and storage — key points

- RAM is volatile working memory.
- ROM is non-volatile memory for stored instructions.

Worked context: Compare primary memory with secondary storage.

Memory, performance and storage — connect the ideas

Follow the concept from left to right

01

RAM is volatile working memory.

→

02

ROM is non-volatile memory for stored instructions.

Apply the principles to this example: Compare primary memory with secondary storage.

Try it · Memory, performance and storage

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

Describe when this principle applies:

ROM is non-volatile memory for stored instructions.

YOUR TASK

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

Teacher check · Memory, performance and storage

Model response · Correct misconceptions

ROM is non-volatile memory for stored instructions.

EXAM TIP / EXTENSION

Use this worked context to add detail: Compare primary memory with secondary storage.

08 Memory, performance and storage · Applications

One clear idea to explain before moving to practice

Cache helps reduce repeated accesses to slower memory.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Memory, performance and storage — key points

- Cache helps reduce repeated accesses to slower memory.
- Clock speed, cores and cache can affect performance.

Worked context: Compare primary memory with secondary storage.

Memory, performance and storage — connect the ideas

Follow the concept from left to right

01

Cache helps reduce repeated accesses to slower memory.

→

02

Clock speed, cores and cache can affect performance.

Apply the principles to this example: Compare primary memory with secondary storage.

Try it · Memory, performance and storage

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

Identify the key technical terms and describe their roles: Clock speed, cores and cache can affect performance.

YOUR TASK

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

Teacher check · Memory, performance and storage

Model response · Correct misconceptions

Clock speed, cores and cache can affect performance.

EXAM TIP / EXTENSION

Use this worked context to add detail: Compare primary memory with secondary storage.

09 Input / output devices · Foundations

One clear idea to explain before moving to practice

Input devices capture data for processing.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Input / output devices — key points

- Input devices capture data for processing.
- Output devices present information or control equipment.

Worked context: Example: temperature sensor → processor → fan actuator.

Input / output devices — connect the ideas

Follow the concept from left to right

01

Input devices capture data for processing.

→

02

Output devices present information or control equipment.

Apply the principles to this example: Example: temperature sensor → processor → fan actuator.

Try it · Input / output devices

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

Compare this concept with a related alternative:
Output devices present information or control equipment.

YOUR TASK

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

Teacher check · Input / output devices

Model response · Correct misconceptions

Output devices present information or control equipment.

EXAM TIP / EXTENSION

Use this worked context to add detail: Example: temperature sensor → processor → fan actuator.

10 Input / output devices · Applications

One clear idea to explain before moving to practice

Sensors provide measurements to systems.

TEACHER EXPLANATION

Discuss why this principle matters in real computer systems.

Input / output devices — key points

- Sensors provide measurements to systems.
- Actuators carry out physical actions.

Worked context: Example: temperature sensor → processor → fan actuator.

Input / output devices — connect the ideas

Follow the concept from left to right

01

Sensors provide measurements to systems.

→

02

Actuators carry out physical actions.

Apply the principles to this example: Example: temperature sensor → processor → fan actuator.

Try it · Input / output devices

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

Apply this idea to a new computer science scenario: Actuators carry out physical actions.

YOUR TASK

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

Teacher check · Input / output devices

Model response · Correct misconceptions

Actuators carry out physical actions.

EXAM TIP / EXTENSION

Use this worked context to add detail: Example: temperature sensor → processor → fan actuator.

Five essential ideas

Recall these without looking at the teaching slides

01 CPU architecture

02 Registers to remember

03 Fetch–decode–execute cycle

04 Memory, performance and storage

05 Input / output devices

End-of-chapter challenge

Answer in your notebook, then compare with a partner

A CPU needs the address of the next instruction. Which register is used?

TEACHER MODEL ANSWER

Program Counter (PC).