



Oxford
International
Lower Secondary

9

Computing

Student Book



OXFORD



Oxford
International
Lower Secondary

9

Computing

Student Book



Alison Page
Diane Levine
Howard Lincoln
Karl Held

OXFORD

Contents

Introduction	3	4 Programming: The fish pond plan	88
1 The nature of technology: Inside the CPU	4	4.1 A model pond	90
1.1 Central processing unit	6	4.2 Plan and make a program	94
1.2 The fetch-execute cycle	10	4.3 Filling the pond	98
1.3 The CPU and logic	14	4.4 Evaporation and rainfall	102
1.4 Complex logical statements	18	4.5 The pond through the year	106
1.5 Logic gates	22	4.6 Warnings and advice	110
1.6 Robots and robotics	26	Check what you know	114
Check what you know	30	5 Multimedia: Creating a multimedia news site	116
2 Digital literacy: Dilemmas	32	5.1 Creating a multimedia platform	118
2.1 Who am I?	34	5.2 Creating news stories	122
2.2 Types of social media	38	5.3 Plan and record a school podcast	126
2.3 Your digital footprint	42	5.4 Edit and publish your audio content	130
2.4 Digital privacy	46	5.5 Plan and create video content	134
2.5 Ethics of care	50	5.6 Edit and publish your video	138
2.6 Healthy balance	54	Check what you know	142
Check what you know	58	6 Numbers and data: Managing projects	144
3 Computational thinking: Artificial intelligence	60	6.1 What is a project?	146
3.1 Antarctic base	62	6.2 Planning a project	150
3.2 Heuristics	66	6.3 Creating requirements	154
3.3 Expert systems	70	6.4 Planning a project timetable	158
3.4 Automate a decision tree	74	6.5 Working on an agile project	162
3.5 Machine learning	78	6.6 Testing software	166
3.6 Training the computer	82	Check what you know	170
Check what you know	86	Glossary	172

Introduction

Delivering computing to young learners

Oxford International Primary and Lower Secondary Computing is a complete syllabus for computing education for ages 5–14 (Years 1–9). By following the program of learning set out in this series, teachers can feel reassured that their students have access to the computing skills and understanding that they need for their future education.

Find out more at:

www.oxfordsecondary.com/computing.

Structure of the book

This book is divided into six units, for Year 9 (ages 13–14).

- 1 The nature of technology:** What the processor is and how it works.
- 2 Digital literacy:** How to participate safely in social media.
- 3 Computational thinking:** The principles that underpin artificial intelligence (AI).
- 4 Programming:** Using computer programs to model real-life systems and solve problems.
- 5 Multimedia:** Creating a multimedia news site featuring sound and video.
- 6 Numbers and data:** Using software for project management.

What you will find in each unit

- Introduction: An unplugged activity and a class discussion help students to start thinking about the topic.
- Lessons: Six lessons guide students through activity-based learning.
- Check what you know: A test and activities allow you to measure students' progress.

What you will find in the lessons

Although each lesson is unique, they have common features: learning outcomes for each lesson are set out at the start; learning content delivers skills and develops understanding.

 **Activity** Every lesson involves one or more learning activities.

 **Extra challenge** Activities to extend students who are able to do more.

 **Test** A short test of four questions, of progressive difficulty, to check students' understanding of the lesson.

Additional features

You will also find these features throughout the book:

 **Word cloud** The word cloud builds vocabulary by identifying key terms from the unit.

 **Be creative** Suggestions for creative and artistic work.

 **Explore more** Extra tasks that can be taken outside the classroom and into the home.

 **Digital citizen of the future** Advice on using computers responsibly in life.

 **Glossary** Key terms are identified in the text and defined in the glossary at the end.

Assessing student achievement

The final pages in each unit give an opportunity to assess student achievement.

- **Developing:** This acknowledges the achievement of students who find the content challenging but have made progress.
- **Secure:** Students have reached the level set out in the programme for their age group. Most should reach this level.
- **Extended:** This recognises the achievement of students who have developed above-average skills and understanding.

Questions and activities are colour-coded according to achievement level. Self-evaluation advice helps students to check their own progress.

Software to use

We recommend Python for writing programs at this age. For other lessons, teachers can use any suitable software, for example: Microsoft Office; Google Drive software; LibreOffice; any web browser.

Source files

 You will see this symbol on some of the pages.

This means that there are extra files you can access to help with the learning activities. For example, half-completed Python programs or spreadsheet files.

To access the files, click 'Download resources' at: www.oxfordsecondary.com/computing.

Teacher's Guides

For more on these topics, look at the Teacher's Guide that accompanies this book.

1

The nature of technology: Inside the CPU

You will learn

- ▶ about the three important parts of the central processing unit (CPU) and how they work together
- ▶ how computers can solve logic and arithmetic problems
- ▶ how robots are used in the modern world and what technology they use.

In Student Book 4 you learned that the processor (sometimes called the 'microprocessor') is at the centre of every computer system. The processor is responsible for all the work your computer does. It controls everything you see on screen. The processor is made up of millions of microscopic electronic switches.

In this unit you are going to put the processor under a microscope. You will look in more detail at the processor and its three main parts. You will learn how the three parts work together to get work done. Then you will look deeper inside to discover how the microscopic switches work.

In the last part of the unit you will learn about robots. You will learn how developments in robotics have been made possible by improvements in the way processors work.



Learning outcomes: Use or describe simple electronic logic gates (for example, AND, OR and NOT gates); Outline the structure of a processor, its components and how they work together; Describe some technical innovations that enable modern robotics

Did you know?

Colossus was the world's first programmable digital computer. It was switched on in December 1943. Colossus was invented to break enemy codes during the Second World War. The computer weighed 1 ton and filled an entire room. Modern cars might contain as many as 50 microprocessors. Each one of these is many times more powerful than Colossus.

In 2007 a replica of Colossus was built. It competed against modern computers of the day in a competition to break a complicated code. Colossus took 3.5 hours to break the code. The winner, a desktop computer, took 46 seconds.



Unplugged

Work in small groups to play a 'True' or 'False' game. Each group should split into two teams. Team A must think of an object. The object is kept secret from the other team, but Team A must say what type of object they are thinking about. For example, if the secret object is a lion, team A say that they are thinking of an animal.

Team B must work out what the animal is by making statements that can be answered with either 'True' or 'False'. In the lion example, statements could be:

Is a type of dog: False

Has a mane: True

Is a lion: True

Teams set each other challenges. The winner of each round is the team that discovers the secret object in the fewest turns.

Talk about...

The word robot comes from the Czech word *robota*. *Robota* means dull, repetitive work. Robots can do jobs that humans find boring and stressful. They can do these jobs 24 hours a day without making mistakes. Millions of jobs that humans do will be replaced by robots in the years ahead. How can we make sure that the changes robots bring will be positive?

computer system
central processing unit microprocessor
control unit arithmetic and logic unit (ALU)
RAM cache robots
vision guided robotics
embedded processor drone

1.1

Central processing unit

In this lesson

You will learn:

- ▶ what happens inside a processor
- ▶ about the parts of a computer's central processing unit (CPU).

Spiral Back



In Student Book 7 you learned that every file you store and use on a computer is made up of digital data. You learned that the brain of a computer is a microprocessor. Inside the processor are millions of tiny electronic switches. In this unit you will learn more about how the processor carries out its work.

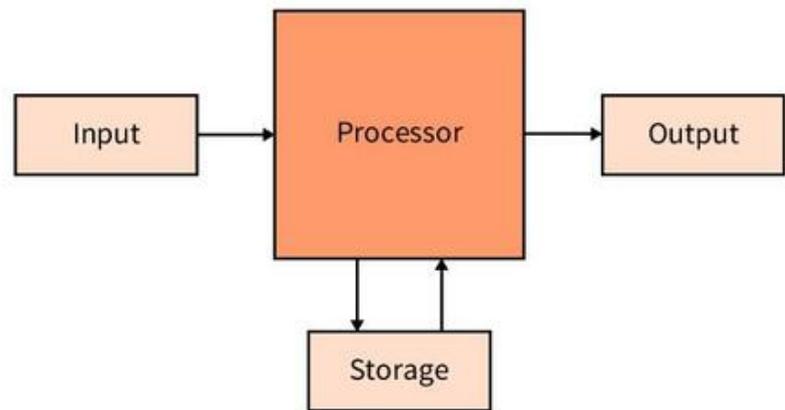
Computer systems

You have gained a lot of experience of using computers on this course and have learned many new and useful skills. Whenever you use a computer you are using a system. A **computer system** is a set of equipment that works together to help you do useful work. A computer system can be drawn as a simple diagram.

A computer system must always have input devices. Input devices allow you to put data into your computer. A keyboard is an input device.

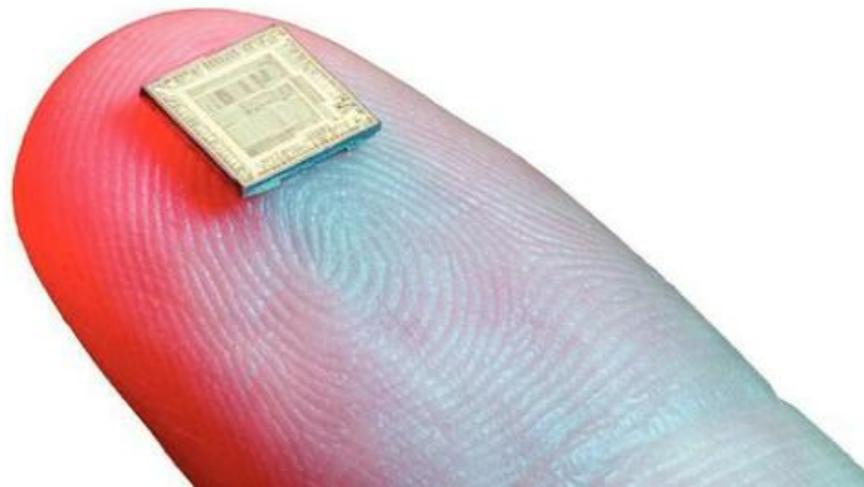
A computer system has output devices. Output devices let you see the results of your work on the computer. A computer screen is an output device.

A computer system has storage devices. You use storage devices to save your work.



The processor

At the centre of your computer system is a processor. In Student Book 4, you learned that a processor does all the work in a computer system. A processor is small enough to fit on your fingertip. Modern processors are so small they are called **microprocessors**.



The central processing unit (CPU)

The **central processing unit** (CPU) is another name for the microprocessor at the centre of your computer system. It is the name to use when you study the computer processor in detail.

The CPU has three important parts: the control unit, the arithmetic and logic unit, and the clock.

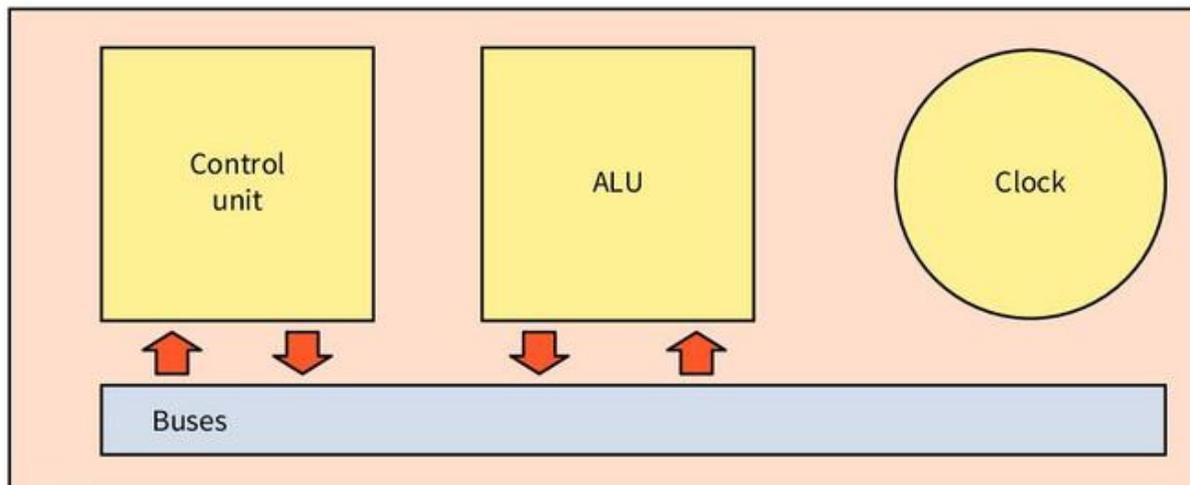
The **control unit** manages the work done by the CPU.

- ▶ When an instruction arrives at the CPU, it goes to the control unit.
- ▶ The control unit works out what the instruction means.
- ▶ The control unit makes sure that the other parts of the CPU do the work needed to carry out the instruction.

The **arithmetic and logic unit (ALU)** does all the calculations in the CPU. If you are working on a maths problem, you might use a spreadsheet to do your calculation. The control unit uses the ALU in the same way. The control unit sends instructions to the ALU. The ALU carries out the instructions.

The **clock** sends out regular electrical pulses just like the tick of a clock. A clock in your home ticks every second. The clock in the CPU of your computer ticks around 3 billion times every second. Every time the CPU clock ticks, the control unit sends an instruction to the ALU.

The CPU



Buses

The three parts of a CPU are joined together by connections called **buses**. Buses are high speed connections that carry data around inside the CPU. They are like the buses you see travelling around towns and cities. Instead of carrying passengers, the buses in a CPU carry data at very high speed.

How the CPU works

Think about the last time you played a game or watched a video on a computer. The screen is full of colour. The images you see are lifelike. Objects move just like they do in the real world.

Movement on-screen is smooth and fast. If you are playing a game, you can give instructions through a joystick or game controller. The action on-screen responds immediately to your command. High-quality audio is being played in the background while you play.



When you experience a game being played on a computer it is easy to think that the CPU must be doing very complicated things. In fact, the CPU can perform only very simple instructions.

For example, a CPU might be asked to add two numbers together with an instruction like 'ADD 2, 3'. Even this simple task has to be broken down into several smaller tasks before the CPU can complete it.

So a CPU can only do very simple tasks. What makes it seem so powerful is that it can do a task every time its clock ticks. The clock in a CPU clicks 3 billion times every second. A computer can appear to do amazing things by doing a lot of very simple tasks, very quickly.

Activity

This activity will give you an idea of how fast a computer CPU does its work. You need two team members for this game, and somebody to time them. Read the instructions and make sure you understand them. Have a practice run.

Start the timer.

- 1 Team member A: Say an action: 'Add', 'Multiply' or 'Subtract'.
- 2 Team member B: Write down the action.
- 3 Team member A: Say a single-digit number (1 to 9).
- 4 Team member B: Write down the number.
- 5 Team member A: Say a single-digit number (1 to 9).
- 6 Team member B: Write down the number.
- 7 Team member A: Tell team member B to work out the answer to the sum.
- 8 Team member B: Work out the answer.
- 9 Team member B: Write down the answer.
- 10 Team member A: Read the answer out loud.

Stop the timer and note how many seconds the task took.

A CPU can carry out the same task 300 million times every second. Multiply the number of seconds it took you to complete the task by 300. That is the number of times (in millions) that a CPU would have done the task in the time it took your team to do it once.

Extra challenge

Three students are working together on the activity. Team member A gives the instructions. Team member B carries out the calculations. Team member C operates the timer. Which part of the CPU does each team member represent?

Test

- 1 What are the three main parts of a CPU?
- 2 How does data move between the parts of a CPU?
- 3 Say one change you could make to a CPU to make a computer work faster.
- 4 Write two places that instructions for the CPU come from.

1.2

The fetch-execute cycle

In this lesson

You will learn:

- ▶ what computer memory is
- ▶ what happens when a computer carries out an instruction.

Memory and the CPU

The CPU is the part of the computer that carries out instructions. You learned in the last lesson that it is made of a control unit, the ALU and a clock, all connected by buses.

The computer's memory is very close to the CPU. It is joined to the CPU by buses. Some people use the word 'processor' to mean the CPU and the memory.

The computer's memory is sometimes called:

- ▶ the memory unit
- ▶ IAS (Immediate Access Store)
- ▶ **RAM (Random Access Memory).**

What is in memory?

Memory holds:

- ▶ the instructions that tell the computer what to do
- ▶ the data values that the computer needs.

In a modern computer the instructions and data are held in the same memory. But they go down two different buses to the CPU.

The memory holds one more thing: the results of the CPU's work.

When the CPU has completed an instruction, it sends the results back to memory.

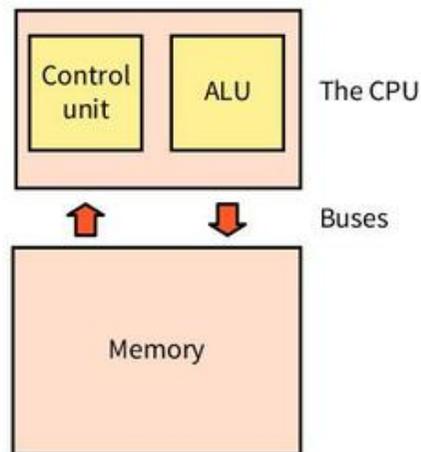
How does memory work?

Memory is made of microscopic electrical circuits. The circuits can be on or off. Everything inside the memory is stored using these on/off signals.

If you completed Unit 1 of Student Book 7, you learned how data is stored inside the memory using on/off signals.

Memory and storage

Data is stored in memory as electrical signals. But if the electricity is turned off, all the data will be lost. This is why you must always save your work before you turn the computer off.



When you save your work, it is copied from memory to storage. Here are some examples of storage:

- ▶ the hard disk of your computer
- ▶ a flash memory drive
- ▶ the storage on your school network
- ▶ cloud storage on the internet.

The important thing about storage is that it stores the data even when the computer is turned off. That means that your work is not lost. Storage is also called **secondary storage**.

Advantages and disadvantages

Both RAM (electronic memory) and storage have advantages and disadvantages.

RAM is very close to the CPU. The CPU can get data and instructions from RAM easily and quickly. The disadvantage of RAM is that its contents are lost when the computer is switched off.

Secondary storage is further away from the CPU. It takes longer for the CPU to get data and instructions from secondary storage than from RAM. But secondary storage has a big advantage – it can keep data and instructions safe when they are not needed, or when the computer is turned off.



Activity

Complete this table to show the advantages and disadvantages of RAM and secondary storage. The first section is done for you.

	RAM	Secondary storage
Advantages	It is close to the CPU. The CPU can get data and instructions from RAM easily and quickly.	
Disadvantages		

The fetch-execute cycle

The CPU carries out instructions millions or even billions of times a second. Every time it carries out an instruction it follows these steps.

- ▶ **Fetch:** The control unit 'fetches' the instruction from RAM. The instruction travels down the bus from RAM to the control unit.
- ▶ **Decode:** The instruction is in the form of a binary number code. The control unit knows all the binary number codes. The control unit 'decodes' the instruction, so it knows what to do.
- ▶ **Execute:** The control unit sends a signal to the ALU to tell it what to do. The ALU carries out the instruction. 'Execute' means carry out an instruction.
- ▶ **Save:** If the instruction produces a result, then the ALU sends the result back to RAM.

These steps are called the **fetch-execute cycle**.

The computer might also need to fetch some data from RAM. Some computers fetch instructions and data in one cycle. Some computers fetch instructions and data in different cycles.

Worked example

In Student Books 7 and 8 you created programs in Python. You will create more programs in this book. Here is a single command written in Python.

```
answer = 2 + 3
```

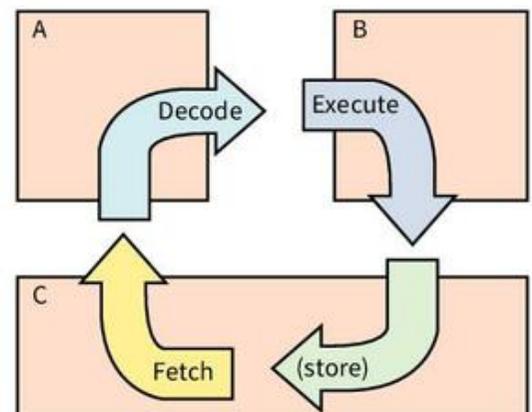
To carry out this instruction the computer must complete at least one fetch-execute cycle.

- ▶ **Fetch:** The control unit fetches the instruction (add) and the data values (2, 3) from RAM.
- ▶ **Decode:** The control unit decodes the instruction and sends a signal to the ALU, telling it to add the numbers together.
- ▶ **Execute:** The ALU carries out the instruction, and adds the two numbers together.
- ▶ **Save:** The ALU sends the result of the addition back to RAM. The result is saved in a memory location with the label 'answer'.

Some computers can do all this in one cycle. Some computers will fetch the instructions and the data in different cycles.

Fetch-execute diagram

You can draw the fetch-execute cycle using a simple diagram like this.



Activity

The parts of the fetch-execute cycle happen in different places:

- ▶ in memory
- ▶ in the control unit
- ▶ in the ALU.

Draw the diagram of the fetch-execute cycle. Instead of letters A, B and C, put the name of the place where each part of the cycle happens.

Memory and computer speed

In the last lesson you learned that the speed of the clock affects the speed of the computer. But the size of memory is also important.

RAM

If a computer has lots of RAM, then all the data and instructions can fit into the memory. The CPU can get the data and instructions very quickly. The computer will go quickly.

If a computer doesn't have very much memory, then the data and instructions won't all fit into memory. Some will have to wait in storage. The computer will go more slowly.

Cache size

The CPU has a small amount of memory which is even closer than RAM. This is called **cache**. It is very quick for the CPU to get data and instructions from cache. If a computer has a big cache then it will be able to get all the data and instructions quickly.

Word size

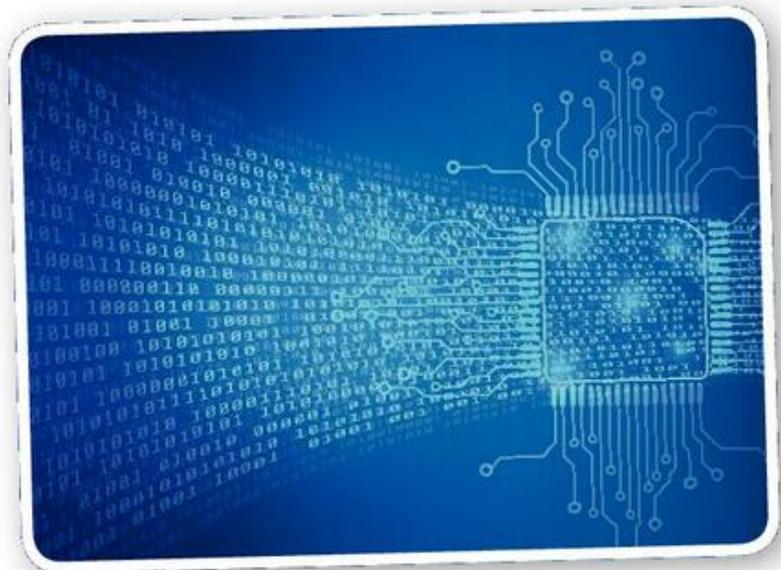
You have seen that some computers can fetch a lot of data from memory in one cycle. Other computers need several cycles. The amount of data that a computer can fetch and use in one cycle is called the 'word size'. A computer with a large word size generally works more quickly. The buses are larger so they can carry more.

➔ Extra challenge

A friend wants to buy a fast computer. Write an email telling them what to look out for when they choose a computer. One factor is clock speed, but there are others. Tell your friend about some other factors that affect the speed of a computer. Explain why each one is important.

✓ Test

- 1 What is the difference between memory and storage?
- 2 List the four stages of the fetch-execute cycle.
- 3 Describe what happens during the 'execute' stage of the fetch-execute cycle, and where it happens.
- 4 Explain why a computer with lots of RAM (memory) will generally go faster than a similar computer with less RAM.



1.3

The CPU and logic

In this lesson

You will learn:

- ▶ how the ALU processes logical problems
- ▶ how to write a logical argument
- ▶ how to draw a truth table.

Arithmetic and logic

You have learned that the CPU contains an arithmetic and logic unit. In the activity in Lesson 1.2 you learned how the ALU can do arithmetic.

If you are playing a game on your computer, you can see the result of the ALU performing arithmetic. For example, the strength of your character increases when you pick up energy during a game. A value is added to your existing strength total.

A game would not be interesting if it only used arithmetic. A game must also include challenges. For example:

- ▶ Does the treasure chest contain gold coins?
- ▶ Does the key open the treasure chest?

Challenges like these cannot be solved using arithmetic. They need logic. In this lesson you will learn what logic is and how it is used by the ALU.



What is logic?

Think about the statement 'the treasure chest contains gold'. There are two possible conditions. The statement can be true or false. 'The key opens the treasure chest' is a logical statement. A logical statement can be used to say if something is true or false.

Activity

'It is raining' is a logical statement. It could be true or false. Write two more logical statements about the weather.

Logic and the ALU

The computer is a digital device. A computer processor is made up of electrical switches. The electrical switches in a computer can be on or off. A computer is called a **two-state** device.

Logic also has two states. The two states are true and false. A logical statement can be true or it can be false. A computer's ALU can process logical statements. It can do so because both logic and the computer use two states.

					
OFF	ON	ON	OFF	ON	OFF
0	1	1	0	1	0

In a computer we use binary to show the state of a switch. A '1' is used to say a switch is 'on'. A '0' says a switch is 'off'. We can also use binary to show the state of a logical statement. A '1' can be used to show a statement is true. A '0' can be used to show it is 'false'.

Linking logical statements

Logic is not just about saying whether statements are true or false. Logical statements can also be used to draw conclusions from data and make decisions. To use logic to draw conclusions you must be able to combine logical statements. The word 'then' is used to combine logical statements.

Here are two logical statements about a computer game:

- ▶ Player has no lives.
- ▶ Game is over.

Each of the two statements can either be true or false. You can link the two logical statements using the word THEN:

- ▶ Player has no lives THEN Game is over.

When two statements are linked, they can be used to draw conclusions. We can say:

- ▶ 'Player has no lives' is true THEN 'Game is over' is also true.
- ▶ 'Player has no lives' is false THEN 'Game is over' is also false.



Activity

Can you think of any occasion when you have used logical statements while learning computing skills? Write down any you can think of. For each, give an example of the logical statements you have used.

Parts of a logical statement

To make it easier to talk about logic, the two parts of a linked statement have names.

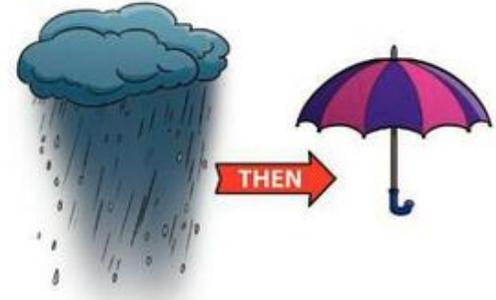
In a logical statement, everything to the left of THEN is called the **proposition**. Everything to the right of THEN is called the **conclusion**.

The whole statement is called a **logical argument**.

Proposition		Conclusion
Player has no lives	THEN	Game is over

Activity

The first activity used 'It is raining' as an example of a logical statement. This statement is used as a proposition in the table below. The conclusion 'Open umbrella' has been linked to the statement. Now we can say that if 'It is raining' is true then 'Open umbrella' is also true.



It is raining	THEN	Open umbrella

In the activity you wrote logical statements about the weather. Copy the table. Write a conclusion to match each of the propositions you wrote.

Truth tables

A **truth table** is a way of laying out a logical statement in table form. It is easier to understand the logic when it is laid out in a table. A written description can be confusing, especially for complex logical statements.

There are four steps in creating a truth table.

1 Write out the argument. Always write THEN in upper-case letters to show that it links the statements:

Player has no lives THEN Game is over

2 Create the column headings. Your table needs a column for each statement in your argument. There are only two statements in this example but there can be more. Always write the conclusion in the last column on the right. There is no need to use THEN in your table.

Player has no lives	Game is over
---------------------	--------------

3 Add a row for every possible response to the proposition. In this example, the proposition is 'Player has no lives'. There can only be two responses: false or true.

Player has no lives	Game is over
FALSE	
TRUE	