

Data handling

**Lessons:** 5

**Programming languages:** MakeCode

**Target age:** 7-11 yrs

**Subjects & topics:**

* Programming: Data handling, Selection, Iteration
* Computational thinking: Pseudocode, Flowcharts
* Mathematics: Information handling

# Unit of work summary

This series of five lessons is aimed at students aged 9-10. Students learn about data through a variety of unplugged activities. They write and evaluate algorithms and programs using selection and repetition to use the micro:bit as a temperature recorder, an automatic warning system and a digital assistant. You will ideally use physical micro:bits for these lessons, although you can also use the simulator.

## Overall key learning

* can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* are responsible, competent, confident and creative users of information and communication technology

## Additional skills

Researching, design thinking, problem-solving, debugging

## Lesson 1: What is data?

Pupils learn about data by researching data about a person of their choosing and exploring ways data can be grouped. They consider the data that might be held about them and look at examples of data misuse by organisations.

**Key learning:**

* To understand what data is
* To classify data
* To identify ways that data might be used

## Lesson 2: Data treasure hunt

Pupils go on a treasure hunt around school to find data before learning about sensors and writing programs to record the temperature in different locations. They consider what the data they have collected shows and identify patterns.

**Key learning:**

* To understand that some devices uses sensors
* To write simple programs using sensors
* To use the BBC micro:bit to collect data

## Lesson 3: Sensor gadget design

Students develop their understanding of sensors through unplugged activities and by writing algorithms using repetition and selection. They then apply their understanding to design and evaluate a gadget using a sensor.

**Key learning:**

* To explain how repetition is used when programming sensors
* To follow design criteria to design a product
* To write algorithms that show how sensors will be used

## Lesson 4: Data conditions & selection

Pupils explore using data collected by the sensors on the micro:bit as a condition in programs. Then plan, program and test using the micro:bit as a temperature warning system.

**Key learning:**

* To know that data can be used as a condition in selection
* To explore the effects of changing the value of data in programs
* To write programs that use data as a condition

## Lesson 5: Digital assistants

Students explore using conditions in selection through unplugged activities before writing a program to enable the BBC micro:bit to be used as a digital assistant.

**Key learning:**

* To read and write algorithms using selection
* To identify how digital assistant might work
* To write a program to use a micro:bit as a digital assistant

# Curriculum links

These lessons are mapped to the following learning objectives and standards for computing, geography, technologies, numeracy and mathematics:

## England National Curriculum

#### KS2 computing curriculum

Curriculum aims:

* can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* are responsible, competent, confident and creative users of information and communication technology

Students should be taught to:

* design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
* use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

[Read the full KS2 computing curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf).

#### KS2 geography curriculum

Students should be taught to:

* use fieldwork to observe, measure, record and present the human and physical features in the local area using a range of methods including digital technologies

[Read the full KS2 geography curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239044/PRIMARY_national_curriculum_-_Geography.pdf).

#### Years 3 & 4 science curriculum

Working scientifically:

* making systematic and careful observations and, where appropriate, taking accurate measurements using standard units, using a range of equipment, including thermometers and data loggers
* using results to draw simple conclusions, make predictions for new values, suggest improvements and raise further questions

[Read the full KS2 science curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/425618/PRIMARY_national_curriculum_-_Science.pdf)

#### KS2 DT curriculum

* technical knowledge - apply their understanding of computing to program, monitor and control their products

[Read the full KS2 DT curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239041/PRIMARY_national_curriculum_-_Design_and_technology.pdf)

## Scotland Curriculum for Excellence

#### Technologies

* I can extend and enhance my knowledge of digital technologies to collect, analyse ideas, relevant information and organise these in an appropriate way (TCH 2-01a)
* I can investigate how product design and development have been influenced by changing lifestyles (TCH 2-05a)
* I understand the operation of a process and its outcome. I can structure related items of information (TCH 2-13a)
* I can explain core programming language concepts in appropriate technical language (TCH 2-14a)
* I can create, develop, and evaluate computing solutions in response to a design challenge (TCH 2-15a)

[Read the full Curriculum for Excellence: technologies](https://education.gov.scot/Documents/Technologies-es-os.pdf).

#### Numeracy and mathematics

* I have carried out investigations and surveys, devising and using a variety of methods to gather information and have worked with others to collate, organise and communicate the results in an appropriate way (MNU 2-20b)
* I can display data in a clear way using a suitable scale, by choosing appropriately from an extended range of tables, charts, diagrams and graphs, making effective use of technology (MTH 2-21a)

[Read the full Curriculum for Excellence: numeracy and mathematics](https://education.gov.scot/Documents/numeracy-maths-eo.pdf).

## Northern Ireland Curriculum - Primary

#### Using ICT across the curriculum

Pupils should be taught to:

* explore - access and manage data and information
* explore - research, select, process and interpret information
* explore - investigate, make predictions and solve problems through interaction with digital tools
* express - create information and multimedia products using a range of assets

#### KS2 - suggested curriculum ideas for the world around us

* design and make models

[Read the full Northern Ireland Curriculum - Primary](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/The%20Northern%20Ireland%20Curriculum%20-%20Primary.pdf)

#### KS1 & 2 - requirements for using ICT

* explore - investigate, make predictions and solve problems through interaction with digital tools

[Read the full KS1 & 2 requirements for using ICT](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Curriculum%20Requirements%20for%20Using%20ICT.pdf)

#### Primary using ICT - desirable features - computational thinking and coding

**Level 4**

Pupils should:

* create a more sophisticated coding project using a broad range of commands; and/or
* solve a given problem using commands in a programming environment.

**Programmable devices (such as Parrot Drone, micro:bit or Sphere)**

* look at and talk about examples of coding projects, including the use of motion, looks, lights or sounds, sensors, control and events such as ‘if...then’ and ‘loop until’ (or equivalent) that make the code more efficient;
* recognise that these projects are composed of different components and break the task into smaller manageable tasks (decomposition);
* in small groups, plan and storyboard their own coding project, working out what different parts of the program must do, using logical reasoning to discuss and compare the commands that are required for their algorithm;
* use a range of commands to create a project including triggering commands such as ‘if...then’ and ‘loop until’ to facilitate a more efficient method of interaction;
* test and debug at regular intervals and collaborate with others to solve problems as they arise;

**Finally**

* share their work (possibly using digital tools), respond to feedback and comment on others’ work; and
* organise files and export work in an appropriate format so that others may view it.

[Read all Primary using ICT desirable features](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Primary%20Using%20ICT%20Desirable%20Features%20Update%202019.pdf)

## Curriculum for Wales

#### Science and technology

Progression step 2 - computation is the foundation for our digital world:

* I can safely use a range of tools, materials and equipment to construct for a variety of reasons
* I can use computational thinking techniques, through unplugged or offline activities
* I can create simple algorithms and am beginning to explain errors
* I can follow instructions to build and control a physical device

Progression step 3 - computation is the foundation for our digital world:

* I can use conditional statements to add control and decision-making to algorithms
* I can identify repeating patterns and use loops to make my algorithms more concise
* I can explain and debug algorithms
* I can use sensors and actuators in systems that gather and process data about the systems’ environment

[Read the full science and technology curriculum](https://hwb.gov.wales/curriculum-for-wales/science-and-technology/descriptions-of-learning/)

#### Digital competence framework

Progression step 1 - data and computational thinking - problem-solving and modelling:

* I can recognise and follow instructions in the appropriate order to perform a task.
* I can organise, select and use simple language to give instructions to others.
* I can control devices giving instructions.
* I can identify errors in simple sets of instructions (algorithm).

Progression step 2 - data and computational thinking - problem-solving and modelling:

* I can detect and correct mistakes which cause instructions (a solution) to fail (debug).
* I can create and record verbal, written and symbolic instructions to test ideas, e.g. the order of waking up through a diagram or flowchart.
* I can change instructions to achieve a different outcome.

Progression step 3 - data and computational thinking - problem-solving and modelling:

* I can understand the importance of the order of statements within algorithms.

Progression step 1 - data and computational thinking – data information literacy:

* I can collect data found in my environment.
* I can sort and classify objects using one criterion.
* I can present and evaluate my data by creating simple charts, e.g. pictogram.

Progression step 2 - data and computational thinking – data information literacy:

* I can collect, enter, organise and analyse data into different groups or formats, e.g. tables, charts, databases and spreadsheets.
* I can extract and evaluate information from tables and graphs to answer questions.

Progression step 1 - producing – creating digital content:

* I can create simple digital work.

[Read the digital competence framework](https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/digital-competence-framework)

## USA Code.org

#### CS Fundamentals

Courses C, D, E

Concepts included:

* data
* algorithms & programs using nested loops & conditionals (if/else if)
* sensors
* variables (strings)

[Read the full Code.org CS Fundamentals curriculum](https://code.org/educate/curriculum/elementary-school).

## USA CSTA Standards

#### Grades 3-5

* 1B-CS-01 - Describe how internal and external parts of computing devices function to form a system.
* 1B-CS-02 - Model how computer hardware and software work together as a system to accomplish tasks
* 1B-CS-03 - Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.
* 1B-DA-06 - Organize and present collected data visually to highlight relationships and support a claim.
* 1B-DA-07 - Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.
* 1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate
* 1B-AP-09 - Create programs that use variables to store and modify data.
* 1B-AP-10 - Create programs that include sequences, events, loops, and conditionals.
* 1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
* 1B-AP-12 - Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
* 1B-AP-13 - Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
* 1B-AP-14 - Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.
* 1B-AP-15 - Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
* 1B-AP-17 - Describe choices made during program development using code comments, presentations, and demonstrations.

[Read the CSTA Standards in full](https://csteachers.org/k12standards/ ).

This content is published under a [Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)](https://creativecommons.org/licenses/by-sa/4.0/) licence.