

Introduction to cyber security

**Lessons:** 3

**Programming languages:** MakeCode

**Target age:** 11-14 yrs

**Subjects & topics:**

* Safety & security: Malware, Ethical hacking, Passwords
* Computational thinking: Algorithms, Pseudocode
* Programming: Variables, Selection

# Unit of work summary

In this series of three lessons aimed at students in the first year of secondary school, students gain a greater understanding of the importance of cyber security and explore the need to create strong password before writing algorithms and programs to create their own ‘strong password generator’ using the micro:bit. Ideally, this unit should be taught after [Computing fundamentals](https://microbit.org/teach/lessons/computing-fundamentals-unit-of-work/).

## Overall key learning

* can understand and apply the fundamental principles and concepts of computer science
* have repeated practical experience of writing computer programs to solve problems
* are responsible, competent, confident and creative users of information and communication technology.

## Additional skills

Problem-solving, collaboration, creative thinking

## Lesson 1: What is cyber security?

Students learn about the importance of cyber security and explore ethical hacking.

**Key learning:**

* to understand about the importance of cyber security in the world today
* to be able to explain what is meant by the term ‘ethical hacking’
* to understand how to recognise potential malware attacks and how to protect data and devices.

## Lesson 2: Strong passwords

In this lesson students explore the need for strong passwords and design an algorithm to create a strong password generator using the micro:bit.

**Key learning:**

* To understand the need for secure password and what makes a password secure.
* To plan, test and debug an algorithm for a password generator.
* To use selection and variables in an algorithm and explain their use.

## Lesson 3: Making a password generator

In this lesson students use their pseudocode algorithm to code their password generator before testing, debugging and evaluating it.

**Key learning:**

* To follow a pseudocode algorithm to program a password generator using a micro:bit
* To write a program using variables correctly
* To test and debug code to create a working password generator

# Curriculum links

These lessons are mapped to the following learning objectives and standards:

## England National Curriculum

#### KS3 computing curriculum

Computing 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
* can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems

Students should be taught to:

* design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
* use logical reasoning to compare the utility of alternative algorithms for the same problem
* use two or more programming languages, at least one of which is textual, to solve a variety of computational problems
* understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns.

[Read the full KS3 computing curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239067/SECONDARY_national_curriculum_-_Computing.pdf)

## Scotland Curriculum for Excellence

#### Technologies

* I can keep myself safe and secure in online environments and I am aware of the importance and consequences of doing this for myself and others. (TCH 3-03a)
* I can informally compare algorithms for correctness and efficiency. (TCH 4-13b)
* I understand constructs and data structures in a textual programming language (TCH 4-14a)

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

## Northern Ireland Curriculum

#### Science and technology - technology and design - statutory requirements, KS3

* Design – identifying problems; investigating, generating, developing, modelling and evaluating design proposals; giving consideration to form, function and safety;
* Show deeper understanding by thinking critically and flexibly, solving problems and making informed decisions, using Mathematics and ICT where appropriate;
* Demonstrate creativity and initiative when developing ideas and following them through;
* Work effectively with others;

[Read the full technology and design statutory requirements](https://ccea.org.uk/downloads/docs/ccea-asset/General/Statutory%20Requirements%20for%20Technology%20and%20Design%20at%20Key%20Stage%203.pdf)

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

**Level 5**

Pupils should:

* create more sophisticated coding projects using a broad range of commands and more than one platform; and
* solve a more complex problem using commands in a programming environment.

**Programmable devices (such as Parrot Drone, MicroBit or Sphero)**

* as a class look at and talk about examples of coding projects, including using multiple ‘if...then’ and ‘if...then...else’ commands, variables, sensors, events, operators and comparators;
* recognise how they can decompose these projects;
* in small groups, plan their own coding project, demonstrating a clear sense of purpose and audience, showing understanding of abstraction by deciding what details they need to include and what they can leave out, working out what different parts of the program must do and using logical reasoning to discuss and compare the commands that are required for their algorithm and predicting the outcome;
* use a range of commands to create a project, including variables, operators and control statements such as ‘if... then...’ alongside the use of ’if...then...else’ and comparators;
* 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 the work of others evaluating process and outcome; and
* organise files and publish work online (if available) so that others can 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 4 - computation is the foundation for our digital world:

* I can decompose given problems and select appropriate constructs to express solutions in a variety of environments.
* I can select and use data structures that efficiently manage data in algorithms.
* I can plan and implement test strategies to identify errors in programs.

Progression step 5 - computation is the foundation for our digital world:

* I can identify, define and decompose problems, choose appropriate constructs and express solutions in a variety of environments.
* I can test, evaluate and improve a solution in software.

[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 4 - data and computational thinking - problem-solving and modelling:

* I can create a simple model or self-contained algorithm.
* I can identify the different parts of an algorithm to determine their purpose.
* I can detect and correct errors in algorithms.

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

## Code.org

Unit 1

Concepts included:

* problem solving
* inputs and outputs
* storing and processing information

[Read the full Code.org CS Discoveries curriculum](https://studio.code.org/courses/csd-2021)

## USA CSTA Standards

#### Grades 6-8

* 2-NI-05 - Explain how physical and digital security measures protect electronic information.
* 2-NI-06 - Apply multiple methods of encryption to model the secure transmission of information.
* 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
* 2-AP-17 - Systematically test and refine programs using a range of test cases.

[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.