

Health tech

**Lessons:** 5

**Programming languages:** MakeCode

**Target age:** 11-14 yrs

**Subjects & topics:**

* Computational thinking: Algorithms
* Sciences: Health
* Design & technology: Product design

# Unit of work summary

In this series of 5-6 lessons aimed at students in the first year of secondary school, students learn about 'health tech', the use of technology to improve health. They develop and apply their knowledge and understanding of computational thinking and real-life problem-solving by working in teams to create their own prototype health tech innovation.

## Overall key learning

* understand and apply the fundamental principles & concepts of computer science.
* gain practical experience of writing computer programs to solve problems.
* evaluate and apply information technology, including new or unfamiliar technologies analytically to solve problems
* be responsible, competent, confident and creative users of information and communication technology.

## Additional skills

Problem-solving, collaboration, critical thinking, creative thinking, prototyping, presenting, researching

## Lesson 1: Health of the nation

Students learn about health tech then research, evaluate and present real-life examples to develop their understanding of how technology is being used to improve people’s health.

**Key learning:**

* To understand some of the UK’s biggest health challenges
* To be able to explain ‘health tech’
* To be research and evaluate real-life examples of health tech being used to address health challenges

## Lesson 2: Health tech innovations

Students are given a health tech challenge and work in teams to design and prototype a health tech innovation.

**Key learning:**

* To develop ideas for health tech innovation innovations to meet a UK healthcare need
* To understand the importance of prototyping
* To select an innovation idea and begin to develop a prototype

## Lesson 3: Prototyping innovations

Students develop their computational thinking skills and complete their prototype.

**Key learning:**

* To work effectively as a team to develop a prototype for a health tech innovation
* To design an accurate, detailed algorithm for at least one prototype feature
* To use the algorithm to write, test and debug a working micro:bit program

## Lesson 4: Preparing presentations

In this lesson, students work in their teams to complete their prototype before preparing and practicing a presentation of their work.

**Key learning:**

* To complete your health tech prototype
* To prepare an effective way to present your prototype
* To practice delivering your presentation

## Lesson 5: Health tech showcase

Students deliver their health tech innovation presentations to an audience, then evaluate their prototype and their learning.

**Key learning:**

* To present your health tech prototype to an audience
* To give and receive constructive feedback
* To evaluate your health tech prototype, presentation and approach to the challenge

# Curriculum links

## England National Curriculum

#### KS3 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
* 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 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems;
* make appropriate use of data structures [for example, lists, tables or arrays];
* understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming;
* create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
* 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 describe in detail the processes used in real world solutions, compare these processes against alternative solutions and justify which is the most appropriate. (TCH 4-13a)
* I can informally compare algorithms for correctness and efficiency. (TCH 4-13b)
* I can explain the overall operation and architecture of a digitally created solution (TCH 4-14b)
* I can select appropriate development tools to design, build, evaluate and refine computing solutions to process and present information whilst making reasoned arguments to justify my decisions. (TCH 4-15a)

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

#### Health and well-being

* I am developing my understanding of the human body and can use this knowledge to maintain and improve my wellbeing and health. (HWB 1-15a, HWB 2-15a, HWB 3-15a & HWB 4-15a)

[Read the full Curriculum for Excellence: health and well-being](https://education.gov.scot/Documents/health-and-wellbeing-eo.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;
* Respond to a personal design challenge in relation to their own lifestyle. Personal understanding
* Explore technical inventions and designs that have met a social need cost-effectively.
* 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;
* communicate effectively in oral, visual (including graphic), written, mathematical and ICT formats showing clear awareness of audience and purpose.

[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 plan and implement test strategies to identify errors in programs.
* I can select and use multiple sensors and actuators that allow computer systems to interact with the world around them.

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 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

Unit 4

Concepts included:

* social impact of computing
* understanding the needs of others when designing a solution
* team project
* testing and acting on feedback
* iteration

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

## USA CSTA Standards

#### Grades 6-8

2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

[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.