

Being active

**Activities:** 3

**Programming languages:** MakeCode

**Target age:** 7-11 yrs, 11-14 yrs

**Subjects & topics:**

* Computational thinking: Algorithms
* Computer systems: Hardware & software, Sensors, Input/output
* Programming: Selection, Sequence
* Physical education: Performance tools
* Digital arts: Wearables
* Global Goals: 3 Health
* Design & technology: Product design

# Design challenge summary

In the **Fitness friend** activity students create a simple wearable device to give regular reminders to do some exercise.

In the **Heart rate monitor** activity students learn how to measure their heart rate and create a prototype of a heart rate monitor.

In the **Walking for water** activity students learn how some children have a daily walk for water and create a step counter to track their steps.

## Overall key learning

* To understand the importance of regular exercise to help prevent heart disease
* To consider how wearable devices can support people to exercise and track their activity level
* To design, code and test a prototype of a wearable device to remind people to exercise using the BBC micro:bit
* 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
* are responsible, competent, confident and creative users of information and communication technology

## Additional skills

Creative thinking, prototyping, testing, debugging, collaboration, evaluation

## Activity 1: Fitness friend

In this activity, students learn about the importance of regular activity then create a simple wearable device to give regular reminders to exercise.

**Key learning:**

* To understand the importance of regular exercise to help prevent heart disease
* To write an algorithm for a 'Fitness friend' device to remind someone to exercise
* To program, create and test a 'Fitness friend' device using the BBC micro:bit

## Activity 2: Heart rate monitor

In this activity students learn how to measure their heart rate and create a prototype of a heart rate monitor.

**Key learning:**

* To understand the importance of activity to help keep the heart healthy
* To measure the effect of different activities on heart rate
* To create, test and evaluate a prototype heart rate monitor using the micro:bit

## Activity 3: Walking for water

In this activity, students learn how some children have a daily walk for water. They then create a step counter to track their own steps.

**Key learning:**

* To develop empathy and understanding for others
* To design and program a step counter using micro:bit
* To use the step counter and undertake a ‘walking for water’ challenge

# Curriculum links

## England National Curriculum

#### KS2 computing curriculum

Curriculum aims:

* 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
* 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 sequence, selection, and repetition in programs; work with variables and various forms of input and output
* use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
* select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

[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 DT curriculum

* generate, develop, model and communicate their ideas through discussion, annotated sketches, cross-sectional and exploded diagrams, prototypes, pattern pieces and computer-aided design
* evaluate their ideas and products against their own design criteria and consider the views of others to improve their work
* 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)

#### KS3 computing curriculum

* design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
* understand several key algorithms that reflect computational thinking (for example, ones for sorting and searching); use logical reasoning to compare the utility of alternative algorithms for the same problem
* understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems

[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 explore and experiment with digital technologies and can use what I learn to support and enhance my learning in different contexts (TCH 1-01a)
* I can explore and comment on processes in the world around me making use of core computational thinking concepts and can organise information in a logical way (TCH 1-13a)
* I understand the instructions of a visual programming language and can predict the outcome of a program written using the language (TCH 1-14a)
* I can demonstrate a range of basic problem solving skills by building simple programs to carry out a given task, using an appropriate language (TCH 1-15a)
* I can design and construct models and explain my solutions (TCH 1-09a)
* I can explore and experiment with sketching, manually or digitally, to represent ideas in different learning contexts (TCH 1-11a)
* 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 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)
* 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)

[Read the full technologies curriculum](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)
* I am aware of the role physical activity plays in keeping me healthy and know that I also need to sleep and rest, to look after my body (HWB 1-27a)
* I can explain why I need to be active on a daily basis to maintain good health and try to achieve a good balance of sleep, rest and physical activity. (HWB 2-27a & 3-27a)

[Read the full health and well-being curriculum](https://education.gov.scot/Documents/health-and-wellbeing-eo.pdf)

## Northern Ireland Curriculum

#### Primary

#### Using ICT across the curriculum

* explore - investigate, make predictions and solve problems through interaction with digital tools
* evaluate - talk about, review and make improvements to work, reflecting on the process and outcome

#### KS1 - suggested curriculum ideas for the world around us

* design and make simple models
* how we grow, move and use our senses, including similarities and differences between ourselves and other children

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

* design and make simple models
* the effects of a lack of basic resources on a place and on people’s lives

[Read the full Northern Ireland curriculum - primary](https://ccea.org.uk/learning-resources/northern-ireland-curriculum-primary)

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

* explore - investigate, make predictions and solve problems through interaction with digital tools
* evaluate - talk about, review and make improvements to work, reflecting on the process and outcome and consider the sources and resources used, including safety, reliability and acceptability

[Read the full KS1 & KS2 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.

**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)

#### Secondary

#### 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;
* explore technical inventions and designs that have met a social need cost-effectively.
* design cost effective and appropriate solutions to meet the specific needs of diverse local and global groups. Citizenship
* demonstrate practical skills in the safe use of a range of tools, machines and equipment;
* show deeper understanding by thinking critically and flexibly, solving problems and making informed decisions, using Mathematics and ICT where appropriate;

[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)

#### Digital skills curriculum

**Become a digital maker at KS3**

* Design a digital solution for a problem using an appropriate method;
* Build a solution based on their design using appropriate tools and techniques;
* Review or test the solution against their original plan;
* Evaluate their solution;

**Becoming a digital worker at KS3**

* Use applications to create products with thought given to both the audience and the purpose through the use of digital design;
* Troubleshoot basic problems with their digital technology.

[Read the full digital skills curriculum](https://ccea.org.uk/learning-resources/digital-skills-hub/key-stage-3-digital-skills-curriculum)

## Curriculum for Wales

#### Science and technology

Progression step 2 - computation is the foundation of 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 algorithms to determine their purpose and predict outcomes
* I can follow instructions to build and control a physical device

Progression step 3 - computation is the foundation of 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

Progression step 4 - computation is the foundation of 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.

[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.
* I can identify repetitions or loops in a sequence, e.g. identify where to shorten a set of instructions by repeating steps, for instance when learning a new song.

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

* I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values and formulae.
* I can understand the importance of the order of statements within algorithms.

Progression step 1 - producing - evaluating and improving digital content:

* I can comment on work in relation to a single success criterion.

Progression step 2 - producing - evaluating and improving digital content:

* I can give an opinion about my own work and suggest improvements based on the success criteria.

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

## USA Code.org

**Computing Fundamentals**

Courses C and D

Concepts included:

* sequencing
* writing algorithms
* writing programs
* debugging
* loops
* events
* programming in a team

[Read the full CS Fundamentals curriculum](https://code.org/educate/curriculum/csf)

**CS Discoveries**

Units 1, 4 and 6

Concepts included:

* iteration
* testing
* problem solving
* team project
* inputs and outputs
* sensors

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

## 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-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-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.

#### Grades 6-8

* 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data.
* 2-CS-03 - Systematically identify and fix problems with computing devices and their components.
* 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
* 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.