**Computing fundamentals**

**Lesson 5: Computer systems 1**

**Introduction**

In this lesson students develop their understanding of computer systems, learning about input and output devices, hardware and software and applying their understanding when creating algorithms and programming using micro:bit.

**Learning objectives**

* To know and understand the common features of computer systems
* To be able to explain input and output devices, hardware and software
* To apply understanding to writing algorithms and programming micro:bit

**You will need**

Lesson plan, lesson guide, printed copies of slide 2 (A3 size micro:bit pictures) for small teams, printed copies of slide 3 labels, scissors, sticky tack, quiz sheets, pens and paper.

**Lesson summary**

1. Name the parts (15 minutes)
2. Hardware and software (10 minutes)
3. Computer systems (10 minutes)
4. Writing algorithms and programs (20 minutes)
5. Wrap up (5 minutes)

**1. Name the parts (15 minutes)**

* Split students into small teams and give out the A3 copies of **slide 2** (micro:bit pictures), printed copies of the labels on **slide 3**, scissors and some sticky tack to each team. Ensure teams have access to micro:bit make code editor and/or a physical micro:bit.
* Explain their task is to stick the labels in the correct places. Highlight they might not know all of them, however they can ‘tinker’ with micro:bit (and use this page if you wish). (NB: more recent micro:bits have a combined compass and accelerometer chip but they are still both separately labelled on the back of the micro:bit).
* Give 5 or so minutes to complete the task, encouraging students who finish to use <https://microbit.org/> to increase their understanding about each part before checking answers as a class (**slide 4**) and encouraging students to share what else they know about (e.g. light and temperature sensors, what pins can be used for etc.)

**2. Hardware and software (10 minutes)**

* Share the learning objectives on **slide 5** if you wish and give out the quiz sheets to teams.
* Explain that micro:bit and the parts they have labelled are examples of computer hardware (**slide 6**) and ask them to complete question 1 before clicking to reveal the definition and discussing as a class.
* Click to reveal each ‘what am I?’ clue on **slide 7** (and answer) for question 2.
* Give teams a timed 1 minute to complete question 3 on their sheet, before checking and discussing as a class.
* Show **slide 8** and ask teams to complete questions 4 & 5 on their sheet (click to reveal answers), awarding a bonus point for any teams who can say what the program shown will do (it is an example of a die).
* Give teams a timed 1 minute to complete question 6, before discussing as a class to assess understanding.

**3. Computer systems (10 minutes)**

* Use **slides 9, 10 and 11** to explain the input-process-output model of all computer systems to students, using the example to illustrate this using micro:bit and highlighting the processor labelled in the introduction task.
* Give teams 2 minutes to label the micro:bit inputs and outputs on their picture sheet and check as a class.
* Ask teams to complete quiz question 7 and 8 to apply their understanding beyond micro:bit before checking as a class.
* If you wish, get teams to ‘mark’ their score for the quiz and award prizes.

**4. Writing algorithms and programs (20 minutes)**

* Give out paper and ask teams them to design a simple algorithm using (at least) one of the micro:bit input and output devices (**slide 12**). You may wish to give some more structure (see support).
* Invite them to swap their algorithm with another team and follow the algorithm they are given to write the program, testing and debugging as they go before feeding back to the other team.
* Share any learnings as a class, focusing on inviting examples of the input and output devices to assess understanding.

**5. Wrap up (5 minutes)**

* Use **slide 13** to assess students’ understanding in their teams and discuss as a class.
* Review the learning objectives on **slide 14** if you wish.

**Extension ideas:**

* You could extend learning to include the CPU and fetch/execute cycle to extend understanding.
* Students could keep a list of all the computers, input and output devices, hardware and software they use in the time before the next lesson to apply their understanding to the real world.

**Differentiation**

**Support:**

* Students can label the parts that are easiest to locate and focus on one or two inputs/outputs only to help support understanding.
* You may wish to give students an example algorithm (e.g. when button A is pressed, make an LED flash). They could also sequence pre-printed instructions if helpful.

**Stretch & challenge:**

* Ask additional questions to stretch and challenge during the quiz.
* Students can be encouraged to create more complex algorithms and programs that make use of repetition and selection and/or multiple input and output devices.

**Opportunities for assessment:**

* Informal observation and assessment of students’ responses during team activities.
* More formal observation of team’s sheets, algorithms and programs if wished.