

# Running CPD sessions



Welcome to *QuickStart Computing*

This section provides guidance about delivering computing CPD sessions. This guidance can be used in conjunction with the two PowerPoint files provided on the CD-ROM that accompanies the toolkit.

## Preparation

### Advertising

Advertise to local primary schools that you are offering two computing training sessions as part of the rollout of *QuickStart Computing*. Use emails and social media (e.g. Twitter), and your own personal contacts at schools as well as local authorities.

### Location

A local school which will allow the use of their computing facilities is ideal. Failing that, any room can be used as long as you have access to a laptop, internet connection and data projector.

### Refreshments

Ensure refreshments are provided for all your attendees.

### Timings

The outlines provided here assume two, three- to four-hour CPD sessions, that would be held during INSET; one at the start of a school term and the second at the end. These sessions could be adapted to fit into three twilight sessions after school.

### Charging

The materials are free and must not be charged for.

### Adapting content

Adapt the resources provided here for use with your audience. For example, you may wish to:

- deliver three CPD sessions, rather than two
- focus on specific content, as requested by your group
- edit the PowerPoint presentations by adding your own content, or deleting some of what's there
- swap the order of sections within the CPD sessions.






It is important to be flexible when running the CPD sessions – be prepared to drop certain activities if attendees would benefit from spending more time on a particular concept.

### Tips and advice

- Try to make sessions as interactive as possible by including hands-on activities or group discussion where all attendees are involved. Avoid long periods of talking from the front; encourage participation by asking questions.
- The best way to learn to code is to code. A great way to learn about the computing curriculum is to plan some lessons and units of work together.
- Share your enthusiasm for the subject.
- Provide suggestions for what teachers should do after the session has finished. Ensure you outline a clear plan for what participants need to do before the next session.

# CPD Session 1

## Resources

-  Welcome to *QuickStart Computing*
-  Session 1 presentation template
-  Skills and knowledge audit form (1 per delegate)
-  Resources audit form (1 per delegate)
-  *QuickStart* handbooks (1 per delegate, either hard copy or pdf files)

## Hardware

- Bee-Bot mat
- Bread, butter, jam, knives, plates, napkins (enough to make a jam sandwich)
- Laptop per delegate with access to Scratch 2.0 online (or Scratch 2.0 installed). Ensure you request this when sending invites
- Data projector
- Internet access and wifi information
- Bee-Bot per group (ensure you request this when sending invites)
- Four-way extension cables for laptops are useful.

## Session outline

### Welcome (5 mins)

- Brief welcome
- Wifi access information
- Structure of the CPD session.

### Why are we here? (10 mins)

- Play the Welcome to *QuickStart* video, or give a general introduction to the *QuickStart* content (see User guide on page 4 of this handbook). It's worth emphasising that computing is more about computational thinking and creativity than 'coding'. Explain how computing is made up of computer science, IT and digital literacy.
- Time permitting, you may wish to play Simon Peyton Jones' TedEx talk about teaching creative computer science: [www.youtube.com/watch?v=Ia55clAtdMs](http://www.youtube.com/watch?v=Ia55clAtdMs).

### How CPD works (5 mins)

- Refer to the *QuickStart* roadmap (see page 5 of this handbook).

### Subject knowledge and skills audit (10 mins)

- Ask attendees to fill in a knowledge and skills audit form. You may find you need to explain some of the vocabulary here.
- Discuss which PoS statements attendees have least confidence with.

### Computational thinking (60 mins)

#### Introduction (10 mins)

- With examples, discuss and define the key computational thinking concepts and processes:
  - logical reasoning (predicting and analysing)
  - algorithms (making steps and rules)
  - decomposition (breaking down into parts)
  - patterns and generalisation (spotting and using similarities)
  - abstraction (managing complexity).

#### Activity: A jam sandwich algorithm (25 mins)

(See Classroom activity ideas on page 8 of this handbook.)

- Explain that in this session, attendees will be writing an algorithm to make a jam sandwich.
- You may wish to play the Introducing algorithms video from the CD-ROM.
- Ask teachers to split up into pairs and write an algorithm for creating a jam sandwich.
- Bring the group back together. Ask for a volunteer to come to the front to make a jam sandwich, following an algorithm provided by the group.
- You can play Phil Bagge's outtake video if you wish: [www.youtube.com/watch?v=leBEFaVHlIE](http://www.youtube.com/watch?v=leBEFaVHlIE)
- Discuss what pupils might learn through such an activity.

#### Activity: Guess my number (25 mins)

(See Classroom activity ideas on pages 11–12 of the handbook.)

- Explain the game to the group (see pages 11–12 of this handbook).
- Split attendees into groups and ask them to come up with the most efficient way of guessing a number from 1–1000, only asking questions that can be answered with a 'yes' or 'no' answer.
- Bring the group back together and ask one member of each group to explain their method.
- Test one or more of these methods, to see how efficient they are.
- Summarise the different algorithms that could

be used to guess the number (random search, sequential search and binary search) and emphasise how a binary search is the most efficient algorithm.

- You can demonstrate this as a Scratch program: <http://scratch.mit.edu/projects/12976768/#editor>.
- Discuss what other problems this algorithm might be used for (e.g. finding words in a dictionary, or books in a library).
- Discuss what pupils might learn from this activity.

### **BREAK (15 mins)**

## **Programming (60 mins)**

### **Introduction (10 mins)**

- You may wish to play the Introduction to programming video from the CD-ROM.
- Explain the relationship between programming, algorithms and code. Emphasise that computing includes both, and that we see programming as the best way to develop and practise computational thinking.
- Discuss and define the key programming concepts of sequence, repetition, selection and variables that are mentioned in the PoS.

### **Activity: Programming a Bee-Bot (25 mins)**

Summary: Some simple programming exercises using a Bee-Bot. (See Classroom activity ideas on page 21 of this handbook for examples.)

- Split attendees into pairs/groups and ask them to write a program to move the Bee-Bot from one position to another (depending on the Bee-Bot mats available). Ask them to record their algorithm, predict what will happen, test and then debug as necessary.
- Provide instructions for a Bee-Bot that include an error. Ask attendees to debug the program.
- Bring the group back together and ask one pair to highlight the bug and how they fixed it.
- You can use the Scratch Bee-Bot simulator at <http://scratch.mit.edu/projects/20050141/#editor> if you wish.
- Discuss what pupils might learn through an activity like this.

### **Activity: Creating 'crystal flowers' in Scratch (25 mins)**

- Ask participants to work out the algorithm and then the program to draw a square.
- Ask them to work out the algorithm and then the program to draw a more complex shape.

- Show attendees a Scratch script for creating a crystal flower, such as <http://scratch.mit.edu/projects/39995570/#editor>. Explain what each part of the script does (reference repetition and variables).
- You can show the Python equivalent code if you'd like and ask participants to spot the connections, but emphasise that there's no requirement for text-based programming in the primary PoS.
- Ask them to make some changes to this script to see what happens.
- Discuss what pupils might learn through this activity.

## **CPD action plan – breakout session (10 mins)**

- Ask teachers to split into smaller groups of 2–3 and spend some time discussing:
  - how they plan to put into practice what they have learned (i.e. running some of the activities with pupils, creating a 'crystal flower' in Scratch)
  - how they plan to deliver computing CPD to their class teachers when they return to school and what support they will need to do this (for example, discuss with senior management team).
- Ask each group to share their plans.

## **General discussion/questions (15 mins)**

Allow the conversation to flow, but you might like to use the following questions to facilitate a discussion.

- How are teachers using computational thinking in their jobs?
- Can the group think of ways in which computational thinking is used across the curriculum?
- What challenges do the group expect with teaching pupils to program Bee-Bots or in Scratch?
- How can teachers best learn programming themselves?

## **Conclusion and close (5 mins)**

- Reminder of follow-up session date.
- Things to do before then:
  - finish knowledge and skills audit form
  - create a CPD action plan
  - start running internal CPD sessions.

# CPD Session 2

## Resources



Session 2 presentation template

- Simple message slips for Activity 1 (see suggested template in diagram on page 34 of handbook)
- Resources for Activity 2: [www.computingschool.org.uk/index.php?id=aberdeen](http://www.computingschool.org.uk/index.php?id=aberdeen)

## Hardware

- Laptop
- Data projector
- Internet access
- Wifi access information
- Enough six-sided dice for all participants.

## Session outline

### Welcome (5 mins)

- Brief welcome
- Wifi access information
- Structure of the CPD session.

### Reflection on school-based CPD (20 mins)

- Ask attendees to split into small groups to reflect on how the CPD has gone so far.
- Bring the group back together and discuss.

### The internet (60 mins)

#### Introduction (10 mins)

- Ask participants to discuss how they'd explain the difference between the internet and the World Wide Web.
- Provide a brief introduction on the internet.
- You may wish to play the How the internet works video from the CD-ROM.

#### Activity: How data is passed between computers (25 mins)

(See Classroom activity ideas on page 34 of this handbook.)

- Explain that the group will be carrying out an 'unplugged' activity to model how the internet passes packets of data.
- Discuss how the activity will work.
- Provide three message slips for each participant.
- Run the activity.
- Summarise the key learning points from the activity. Allow time for participants to ask questions.

#### Activity: How a search engine works (25 mins)

(See Classroom activity ideas on page 41 of this handbook: [www.computingschool.org.uk/index.php?id=aberdeen](http://www.computingschool.org.uk/index.php?id=aberdeen).)

- Explain that this is an 'unplugged' activity to model how a search engine ranks results.
- Summarise how the activity will work.
- Stick the web pages up around the room.
- Run the activity.
- Discuss the key learning points from the activity.

### BREAK (15 mins)

### Planning (20 mins)

- You may wish to play the Planning a scheme of work video from the CD-ROM.
- Discuss options for creating or adapting a computing scheme of work.
- Ask teachers to split out into small groups to plan a computing lesson.
- Ask each group to summarise their plans.

### Teaching (20 mins)

- You may wish to play the Approaches to teaching computing video from the CD-ROM.
- Open up the floor for a general discussion about teaching approaches for computing and encourage individuals to share their experiences.
- If discussion flags, ask about whether participants think teaching computing and teaching ICT are that different?

### Assessment (20 mins)

- You may wish to play the Assessing and tracking progress video from the CD-ROM.
- Provide a few examples of Scratch programs that you (or pupils) have created, or just browse the gallery of uploaded Scratch projects at [http://scratch.mit.edu/explore/?date=this\\_month](http://scratch.mit.edu/explore/?date=this_month).
- Ask participants to split into small groups and use the Progression Pathways, or the Attainment Targets (PoS statements) to assess each program.
- Bring the group back together to share thoughts and ideas.

### General discussion/questions (15 mins)

Use the following questions to facilitate a discussion:

- How would you teach pupils how the internet and search engines work?
- How would you structure your scheme of work?
- What makes a good computing lesson?
- What do you see as the challenges for assessing computing?

### Conclusion and close (5 mins)