Buzzing with Bee-Bots

Background notes for teachers

Years F–2 band

The following learning experiences, lessons and activities have been designed to support students within the years F–2 band who have had no previous engagement with the Australian Curriculum Digital Technologies. They enable students to develop foundation skills in computational thinking and to develop an awareness of personal experiences using digital technologies.

The years F–2 band lesson examples provided in the Digital Technologies Hub support students to learn and apply digital technology knowledge and skills. Students interact with others via guided play and learning experiences integrated across the learning areas.

The lessons provide examples of how learning can be structured and content delivered. The lessons and activities can be modified to suit context and learners' needs. Lessons may be taught in their entirety, or may be broken into separate learning experiences and taught over a number of days. Some lessons contain suggestions and strategies about where ICT can be used effectively to support learning.

How is programming addressed in the F–2 band?

Different programming languages are addressed in the Digital Technologies curriculum. In the years F–2 band, there is no requirement for students to learn a particular programming language as this is the pre-programming stage. In years F–2, students learn basic programming skills such as steps and decisions required to solve simple problems (for example, programming a robotic toy to move in a certain direction). The focus is on designing a sequence of steps. There are also opportunities within other learning areas to undertake pre-programming activities to support curriculum concepts. The Digital Technologies lessons in the Digital Technologies Hub align where possible with curriculum concepts across learning areas.

In the years F–2 band, the pre-programming activities can be delivered as 'unplugged' activities (activities that do not require the use of a computer). Unplugged activities can be undertaken in all bands. In the years F–2 band, unplugged activities give students opportunities to explore concepts such as algorithms and abstraction. Students are engaged through guided play, with hands-on, kinaesthetic and interactive learning experiences. For example, if students are beginning to conceptualise algorithms as a sequence of steps for carrying out instructions, an unplugged activity could have students role-play the steps required to complete a task or solve a problem such as brushing teeth or making toast. Alternatively, students could be asked to arrange a set of image cards in the correct sequence of steps.

What is coding?

The terms programming and coding are often used as if they mean the same thing. However, there is a difference. Coding is the sequence of steps or commands for a computer or other digital device to follow. Coding is the process of translating a program into language that can be understood by a computer. This is done using a formal, constructed language – a programming language – with its own rules, grammar and syntax. Different digital systems understand different programming languages.

Buzzing with Bee-Bots

What is programming?

Programming is a more holistic process of which coding is a part. It includes breaking down the problem and designing the steps and procedures that inform the code. Computer programming is the process of writing a computer program – formulating the problem and taking it from concept to solution.

In the Digital Technologies curriculum, the process of creating a digital solution includes defining, designing, implementing, evaluating, collaborating and managing.

- **Defining** is understanding the problem and breaking it into smaller parts.
- **Designing** is planning what the solution will look like and how it will work. This includes designing a sequence of steps and decisions or algorithms that will be automated.
- **Implementing** is creating and modifying the program that has been designed, then **evaluating** the solutions. While undertaking this process, students are **collaborating**, **managing** and working with others to create the digital solutions, as well as communicating information and ideas.

What is a program?

A (computer) program is an algorithm – or set of algorithms – that a computer can execute to perform behaviour that has been pre-determined by the programmer to solve a solution or meet some other need. It has been coded into something that can be understood by a computer.

What is computational thinking?

Computational thinking is a problem-solving method. It includes a range of strategies such as decomposition (breaking down the problem into smaller tasks); identifying and interpreting patterns in data; organising data in a logical way; and algorithms and abstraction.

Abstraction is about simplifying things. It is used when defining problems, to identify the most important information. This requires simplifying a complex idea, problem or solution by hiding details that are not relevant, enabling focus on a manageable number of aspects. For example, when explaining the process involved in making toast, it is not necessary to describe where the toaster is located, what type of bread is being used, whether the toaster is silver or black or whether the bread is thickly or thinly sliced. A soccer training timetable is an abstraction of what happens in a typical week. It includes key information about who is trained, when, where and by whom. It does not include layers of complexity such as the aims of each session, equipment needed, drills or exercises planned for each session, player names and player injuries. You can find the Australian Curriculum Digital Technologies Glossary here.

Algorithm: An algorithm is a sequence of steps and decisions required to solve a problem. For example, travelling to school can be considered as an algorithm or a sequence of instructions – leave home, turn left, travel for 6 km, turn right. GPS devices such as those in cars use algorithms to work out the fastest way to travel from one location to another. A flowchart is another example of an algorithm where a user follows a set of instructions.

Buzzing with Bee-Bots

Digital technologies curriculum

Within the Australian Curriculum Digital Technologies F–2 band, the content description that relates to algorithms include:

DEFINING AND DESIGNING

Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004)

F–2 useful hardware and software

There is a range of hardware and software to support learning in the Digital Technologies F–2 band. Some of the programmable toys, such as Bee-Bots and Sphero, have visual programming apps available for use in combination with the devices.

Buzzing with Bee-Bots

Bee-Bots are colourful, easy-to-use, programmable robots. They can be used as fun toys to teach simple programming concepts, directional language, sequencing, estimation and problem solving. Students can use the Bee-Bot's arrow keys to instruct it to move forward or backward in 15 cm intervals, or to turn at a 90 degree angle to the left or right. Students simply press the **Go** button to make the Bee-Bot move.

There is also a Bee-Bot app that enables students to use their iPad to program the Bee-Bot through sequences forwards, backwards, left and right at 90 degrees.