

Fitness Unplugged

Computer programs are all around us, in things like fitness devices, smartphones and alarm systems. These programs are referred to as “applications”, “apps” or “software”, and often they are nearly invisible, in devices, software “boots up” when you switch it on and keeps going until you switch it off. Despite this variety, they all work on similar principles that are accessible to students of all ages, and this unit explores some of the fundamental principles in programming.

What is It?

Writing a computer program involves planning what you’re going to do: “coding” the instructions, testing them, tracking down any bugs, and changing the program to that it works correctly. Most substantial programs are written by a team of people, and often the roles of design, coding and testing are separated out. This team is usually comprised of the Programmer (the people who write the program), and the Testers (the people who test the program).

Why?

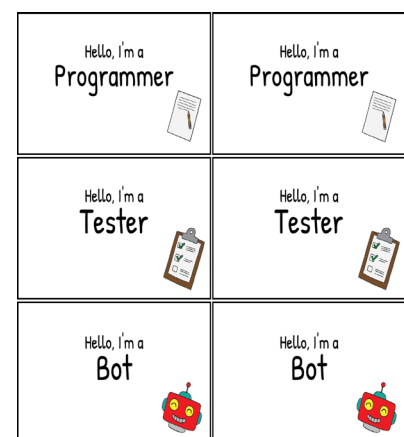
These Kidbots activities separate the programming from the testing to avoid the programmer adjusting their program on the fly, and also support students to understand that programming is about working together, thinking through what you want to have happen and collaborating to solve problems.

Link to Digital Technologies Curriculum

This activity contains most of the components of the first Progress Outcome in **Computational Thinking**. The symbols on the cards represented the physical actions students needed to perform so are an abstract representation. The activity also contains strong elements of algorithmic thinking and debugging.

Mathematics: Numeracy, Geometry
Literacy: Speaking

Downloadable Resources (One Per Class):



Job Badges

Classroom Resources:

Clipboards
Handheld whiteboards
Hula Hoop
Paper
Pens
Whiteboard pens

Activity Background

Today we are going to write our own unplugged fitness app. Before we start we need to design a programming language so that we know what to do.

The Activity

1. Brainstorm as a class all the different fitness exercises you could have in your programming language.

Ideas to get you started include:

- Running on the spot
- Star jumps
- Ski jumps
- Balancing bean bags on your head

2. Work in your groups and draw on each of the cards. You should design:

- A "GO" card to show the start of the program
- 4 cards, each with a different exercise depicted on it
- A card with that depicts how you should finish. Ideas could be:
 - Strike a pose
 - Sit with your arms crossed
 - Stand as straight as you can

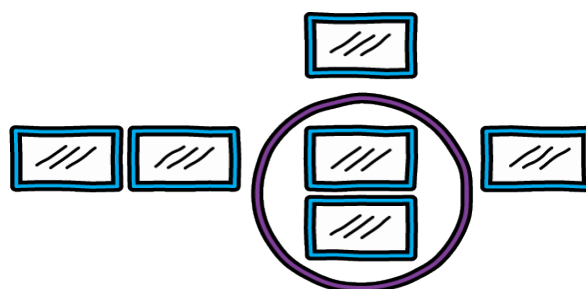
3. Now place your cards in a row with the GO card first, then with each picture and how you should finish last. (There should be no numbers to indicate how many times you should do the exercise - that will come later)

4. Each group goes around and tries to work out what each of the cards mean from the other groups.

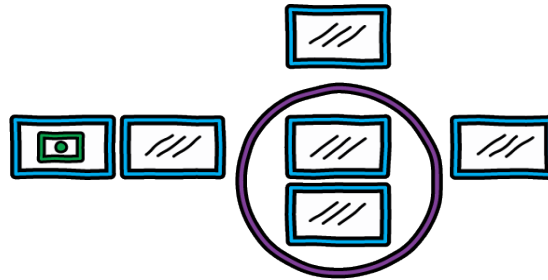
5. Come together and discuss what was the same and for which cards you had to "learn" because you weren't sure what it was?

6. This is the same with computer programming languages. You can learn to program with different languages, and some things seem similar in a new language, while other things you have to learn as it isn't clear how to use a particular command.

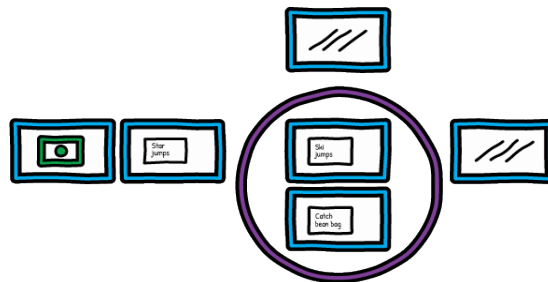
7. Now put out 6 whiteboards and the hula hoop like this:



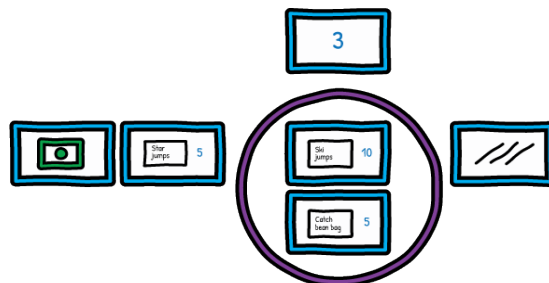
8. Have one group put their Go card on the first whiteboard



9. Now have them put their exercises cards out on the whiteboards shown. and explain that whatever is in the hula hoop will be repeated (it's a loop!)

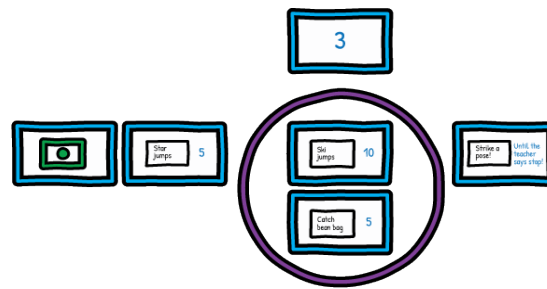


10 We need to set the parameters. A parameter tells us how long to do something or how many times to do something. Here we know to Go, to do star jumps (but not how many), to do ski jumps (but not how many), to catch a bean bag (but not how many) and that we should repeat the ski jumps and catch a bean bag, (but not how many times)!



11. With your whiteboard pens, write in the number of times each exercise should be done. (Encourage the students to choose realistic numbers; the number on the hula hoop multiplies the ones inside.) For the exercises in the hula hoop, you need to repeat both of them how many times? (In the example, the sequence of 10 ski jumps and 5 catches is repeated 3 times, so it would be 10 ski, 5 catch, 10 ski, 5 catch, 10 ski, 5 catch).

12 The last piece of programming is to show how we will end when we have finished our fitness app workout.



13. Now we have built up our fitness app, it's time to test it. The Programmer watches to see if everyone is doing it correctly. The Tester is looking for if there are any numbers to change or if the activities need further instructions. When the Tester says "Go" start following your fitness app.

14. The Programmer asks the Tester if they saw anything they thought could be changed or improved?

Extending The Lesson

Getting the sequence right for programming is very important. If any detail is missing then your program won't run how you expected it. A loop in programming is when you repeat the instructions inside of it, until a certain condition is met (in this case each exercise is completed a certain number of times). A loop can get a lot of instructions to happen with a relatively short program.

- Discuss all the areas where you needed to collaborate together. What skills did you need to apply when discussing what should be programmed.
- Do you think this is a way you could set up fitness for your class?
- Do you have ideas for writing different fitness programs this way?

