Junior Computer Science

code-it.co.uk

Train your computer to do maths

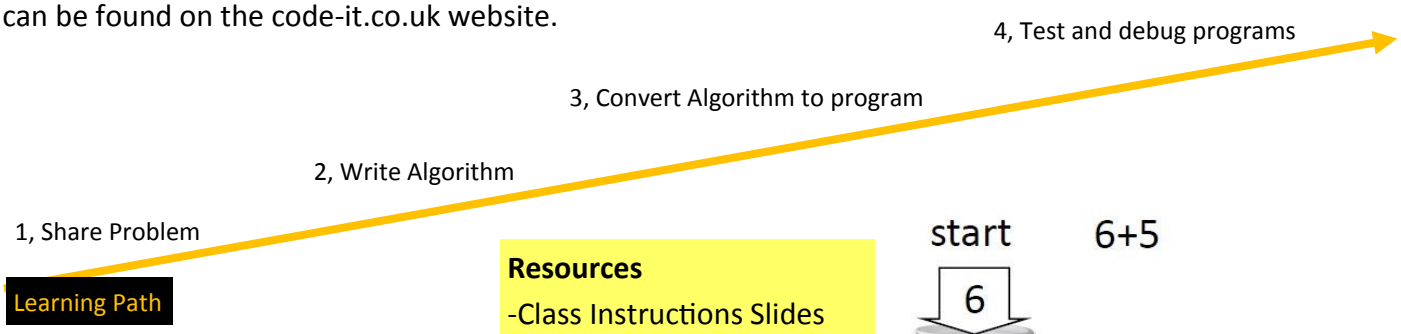
Computer Science Concepts
-Converting user input into a variable
-Writing Algorithm
-Converting Algorithm into program
-Working with multiple variables

Maths Concepts
-add
-subtract
-Divide
-Multiply

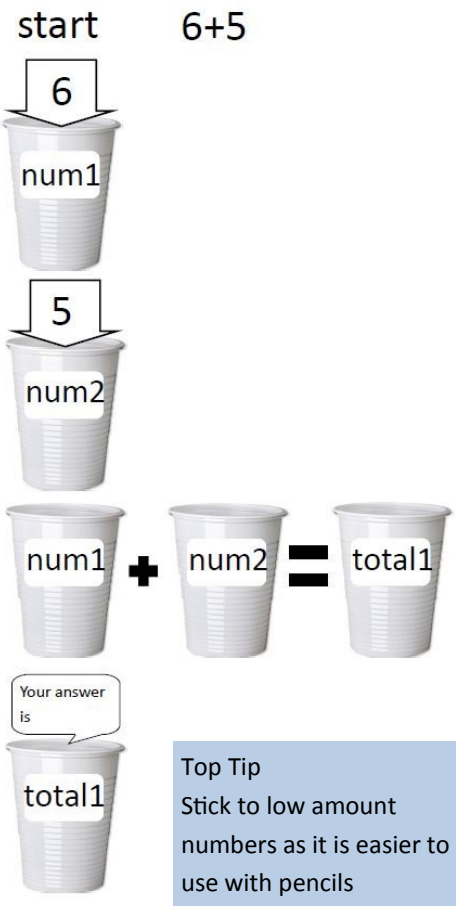
Program Aim Write algorithms and convert them into programs to solve simple maths questions

Computing Program of Study
-design, write and debug programs that accomplish specific goals -work with variables

Differentiation and Assessment for Learning At the beginning of each session the *learning intention sheet* is shared and the learning journey expanded through success criteria. Pupils feed their progress back to the teacher through annotating this sheet with smiley faces at the end of the lesson. Teachers can also annotate the sheet to indicate those who need more or less help in future lessons. These extra resources can be found on the code-it.co.uk website.



Resources
-Class Instructions Slides
-Plastic beakers 4 per pair
-Beaker cards cut up
-Lots of colour pencils
-Challenges printed



1, Share Problem
Use the slides to explain how we can train our computer to do maths.
Explaining that computers know nothing unless humans tell them what to do.

2, Write Algorithm
Use slide 11 (class instruction slides) to go over the rules. Give out 4 beakers per pair and a large handful of pencils as well as the beaker cards cut up. Go over the rules especially the one about numbers only being in pots. Use pencils to put into pots as you do. Go over 6 + 5 =? With the whole class modelling this with apparatus and then creating it as to right

Top Tip
Stick to low amount numbers as it is easier to use with pencils

Junior Computer Science

code-it.co.uk

Train your computer to do maths P2

3, Convert Algorithm to Program

Make sure pupils have got their algorithm sheets on them. Briefly recap design process from slide 9 and how pupils have designed sets of instructions to be tested. Explain that we now need to convert their algorithms into computer programs. We are going to use Scratch but we could use lots of other computer programming languages such as Logo or Python to do the same thing. Warn pupils that the computer language could be very different from our written algorithm.

Move onto Slide 13 which converts their algorithms into specific Scratch blocks.

Show pupils how to create variables (pots)

1, Left click on variables
2, Left click on make a variable
3, Name variable num1 etc

Drag out the blocks needed to make each example and demonstrate how they snap together. Now get pupils to work on converting their algorithms into programs. Remind them that the order of their algorithm should stay the same.

4, Test and debug programs

Did they work? If not why not? Don't be afraid to go back to the pots and cards.