UpTIME: A teaching sequence for primary computing

UpTIME is a simple way to plan effective sequences of activities to teach computing. It is particularly appropriate for the computer science strand of computing but can also be applied to IT and digital literacy learning. The aim of the sequence is to create opportunities for children to develop their digital fluency through motivating, creative and challenging activities that maximise opportunities for children to learn through productive talk with peers and adults. The sequence assumes that the teacher has identified a specific aspect of computing that they wish to teach based on prior assessment of pupils' learning. The sequence might be used within a single lesson or a relatively long series of lessons depending on the complexity of the learning intentions.

U/p – USE / play

We want to set an authentic and purposeful context for the computing concepts and skills that we wish pupils to learn. Often an effective way of achieving this is through pupils using programs that make use of the specific elements of computing being taught. If this program is in the form of a game then they should play the game. For example, if the focus of the lesson is concerned with understanding and using variables then you might start by asking the class to play a game that makes use of variables in a number of different ways. Pupils can be encouraged to try to explain to a partner or the class how they think the game works. Sometimes, the initial program that children use or play may be incomplete or 'broken' to draw their attention to the part of the program that has gone wrong.

T – TINKER

The second step is to investigate the particular learning focus by tinkering with the program or game that was used in the first part of the sequence. This might involve looking closely at the code used in the program. Pupils can discuss with their partners what the different parts of the code do and the teacher can guide the pupils to concentrate on the lesson focus. Pupils should be encouraged to tinker: change the code and see what happens. As they do this they should tell their partner what they predict will happen and discuss the outcome. The teacher will often want to share some of these tinkerings with the class and use this to draw out and introduce key concepts and the vocabulary they want to hear pupils using. In our example of teaching variables, the class might change the amount that a variable increases at points in the program or they might investigate the effect of changing the name of a variable (does it matter to the program whether the variable is named 'score' or not?)

I – IMPROVE

As soon as the pupils start tinkering they will find ways to 'break' the program and ways to improve it. For example, they might think that the game that they have been playing is too easy and want to make it harder by adding obstacles or timers. While some ideas for improvement will come from the children, the teacher can also set improvement challenges. To cater for the range of abilities that can be found in any class, we need to set a range of different challenges that allow all children to achieve our learning intentions but also allow children to select the challenge that is most appropriate for them. For our example variables lesson, we could set challenges that involve making small changes to existing variables (e.g. varying the amount a score changes by), adding new ways that an existing variable is used or displayed, or adding a completely new variable (e.g. 'lives').

M – MAKE

Since the earliest days of primary school computing, it has been clear that one of the most powerful ways that children can learn is by constructing. By giving children the opportunity to design and make their own projects, we can develop independence, motivate children to learn new concepts to achieve a specific outcome, and enable children to make connections between abstract concepts and their application. We can help pupils to design programs, decompose them into manageable tasks, create, debug and share their work. When children create programs in groups they will make predictions, explore ideas together, share ideas and suggestions and develop their computational thinking.

E – EVALUATE

Children will evaluate their work at every stage of this teaching sequence. They will evaluate the programs that they explore at the start of the sequence and they will improve and debug their program continually when programming. In addition to this, we need to create opportunities for children to evaluate their own creations. In doing this children, can evaluate the success of their program for its purpose, assess their own skills and identify areas for future learning.

Chris Shelton (2015)