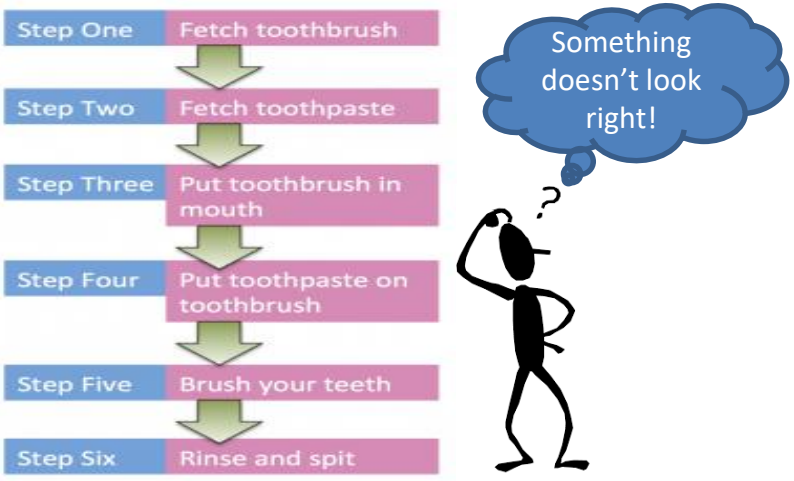
Year 3 – Summer Term 2
We Are Bug Fixers
Computing

In this unit, the children work with different example Scratch projects. They will explain how the scripts work, and find and correct errors in them. The children learn to recognise some common types of programming error, and practise solving problems through logical thinking.

Prior Knowledge

Children will already know how to **program** their own **algorithm** with multiple steps in a logical way.



Computing Key Vocabulary	
Algorithm	A set of rules or a precise step-by-step guide to solve a problem or achieve a particular objective.
Arithmetic Bugs	Errors in a program caused by problems with mathematical calculations.
Bugs	Mistakes in a piece of code.
Conceptual Bugs	Errors in programs owing to a misunderstanding of the system or processes being modelled.
Debug	To fix the errors in a program.
Instruction	An order given to a computer processor by a computer program.
Multi-thread bugs	Errors in programs that involve more than one process running simultaneously. For example, where two processes are waiting for the other to complete, or one process racing ahead of the other
Performance Bugs	Poor functioning of computer programs.
Program	A stored set of instructions in a language understood by the computer.
Script	A script is a set of steps that a programmer writes for a computer to follow.

E-safety: Review
To develop strategies for staying safe when using the internet.

Computer Science

SWITCHEDON Computing

Year 3 – Summer Term 2
We Are Bug Fixers
Computing

This half term we will be using…	
Hardware	**Software/Apps**
Chromebooks	Scratch

Different types of bugs

Off-by-one bug
Here, an instruction in a program repeats one too many, or one too few, times.

Performance bug
This is where a program doesn't perform as well as it could. It could work more quickly or efficiently.

Multi-thread bug
These bugs occur when several things need to happen at the same time.

For example, there might be two processes where each is waiting for the other to complete, or one process that races ahead of the other.

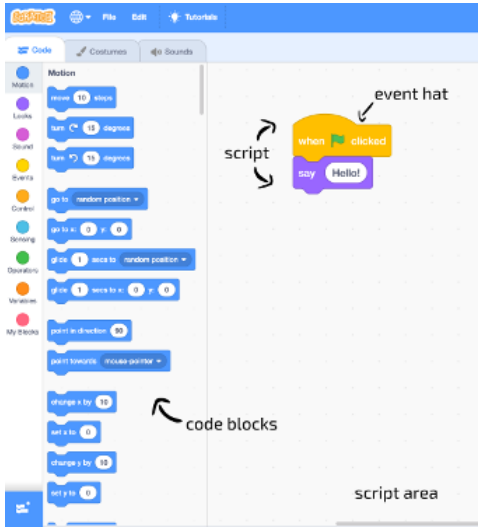
Conceptual bug
This happens when the programmer hasn't fully understood the idea of what is supposed to happen in the program. The bug lies in the idea for the program rather than the code. These sorts of bugs are tricky to find and fix!

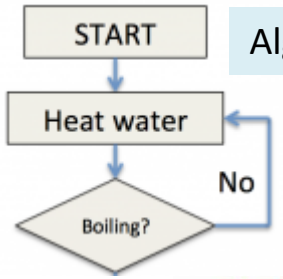
Arithmetical bug
These bugs occur when the computer cannot 'do' the maths required, such as divide by zero.

Resource bug
These bugs happen when the programmer hasn't fully understood how the language or the operating system actually works, so the programming they want to do can't be done.

Computing

RISING STARS





Algorithm