Year Group: 1	Term: Spring 1 & 2	Topic: Computing Science

NC Links

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

Other Curriculum Links

PE – creating obstacle courses/dance, Maths – position and direction, Literacy – instructional writing

Topic Overview

Children will be formally introduced to computing science, learning about algorithms and programming by using a range of activities and applications. Topic will begin by exploring concepts throguh 'unplugged' activities using Barefoot Computing, to support children's understanding of computing science and the vocabulary behind it (use knowledge organisers to help explain concepts and vocabulary). Children will then begin using Code.org (Unit A). Class teachers can also use school ipads as an additional activity or exploration lesson.

Links to Rights Respecting

Article 17 - Every child has the right to reliable information from the media. This should be information that children can understand. Governments must help protect children from materials that could harm them.
Article 28 -Every child has the right to an education. Primary education must be free and different forms of secondary education must be available to every child. Discipline in schools must respect children's dignity and their rights.

Links to North East Ambition

Children will look at different careers within computing and how the subject is evolving every day. Teachers can reference jobs/companies that are recognised globally (apple, Microsoft etc.) or locally (Sunderland Nissan, Newcastle University, CAS). Jobs may include: Engineer, Game Designer, Cyber Crime Officer, Photographer, Video Animator, Office Worker etc.
GATSBY BENCHMARK 3
GATSBY BENCHMARK 4
GATSBY BENCHMARK 5

Possible Visits/Visitors

Essential Subject Skills to be covered

- Control motion by specifying the number of steps to travel, direction and turn.
- Add text strings, show and hide objects and change the features of an object.
- Select sounds and control when they are heard, their duration and volume.
- Control when drawings appear and set the pen colour, size and shape.
- Specify user inputs (such as clicks) to control events.
- Specify the nature of events (such as a single event or a loop).
- Create conditions for actions by waiting for a user input (such as responses to questions like: What is your name?).

Overall Learning Outcomes

Children introduced to computing science and activities/vocabulary such as algorithms and debugging. Allow for plenty of practical opportunities for children to both follow and create algorithms. Explore to children common ways in which algorithms are used in everyday life. Two most common are probably *'route algorithms'* (e.g. using a Sat-Nav) and when cooking *(recipes).* In both examples 'debugging' will often be required. Introduce exploring with children the different number of algorithms (routes) that can be created for even the simplest of algorithms. Which are the most **precise?** (Ones with the shortest number of steps). Children will learn about sequencing, Loops and Events.

Learning Intentions (for use in self assessment at end of topic)

- I can control motion by carefully choosing the number of steps to travel, direction and turn.
- I can add text, show and hide objects and change the features of an object.
- I can select sounds and control when they are heard, how long for and volume.
- I can control when drawings appear and set the pen colour, size and shape.
- I can choose user inputs (such as clicks) to control events.
- I can choose how an events plays (such as a single event or a loop).
- I can create rules for actions by waiting for a user input (such as responses to questions like: What is your name?).

Year Group: 1	Term: Spring 1 & 2	Topic: Computing Science

Possible Activities

- **Barefoot Computing** – BeeBots (Can use BlueBots) basics, BeeBots 1,2,3 programming, Crazy Characters Algorithm
- **Code.org** – (Unit A) Sequencing, Loops and Events
- Plenty of practical opportunities for children to both follow and create algorithms. Children could *'program'* each other to act as robots around the classroom, or could use lines etc. in the hall or playground to program each other around a certain route. P.E. lessons are also great for introducing programming concepts *(algorithms/debugging*) through use of benches, beanbags, hoops etc.
- Think also about how links can be made to topics *(e.g. guiding a Pirate ships around rocks or to the treasure etc.).*
- Use *'real-life'* examples *(e.g. putting on shoes and coats)* to reinforce the notion of algorithms and debugging. As a set of instructions *'finally put on your shoes'* would be acceptable, but as an algorithm *(precise)* more steps would be needed to explain exactly how we put on our shoes.
- Action songs are a great way to also introduce pupils to algorithms. The ***'Hokey Cokey'*** is a great example of an algorithm. The goal – to do the Hokey Cokey. How do we do it? . . . put our left arm, our left out etc . . . Start by simply playing and doing the Hokey Cokey. Progress to pausing and predicting what comes next. End with instrumental version and use cards with different parts of body coloured in. Can pupils follow a newly created algorithm, created Hokey Cokey algorithms of their own *(who says that left arm must come first*), or debug Hokey Cokey algorithms into correct order etc.
- Other action songs such as ***Head/Shoulders, Wind the Bobbin, I'm a Little Teapot*** etc. are also great for the introduction of algorithms and debugging.
- The ***'Code-a-Pillar'*** from Fisher Price and **'BlueBots'** are excellent physical resource to introduce pupils to programming concepts.
- ***'Helicopter Rescue'*** part of Busy Bundle app, or on Busythings website. What is the goal? To get the cow into the bath. How are we going to do this? Fly the *helicopter (algorithm – up, down, left and right). Use of variable (suction pad – on or off).* Allow pupils to experience the activity first, with teacher focusing on terminology of *'algorithms'* and *'debugging'*. Introduce laminated arrow cards. Can pupils construct algorithm as they are performing the activity? *'Logical Reasoning'* – It is logical for us as adults that different coloured cows must go into baths of the same colour, but do all pupils see and recognise this?
- ***'Path Puzzler'*** app also from Busythings. Children to experience creating and debugging of route algorithms.

Suggested Strategies for Recording Learning

- Ongoing projects recorded on ***'Helicopter Rescue' or 'Path Puzzler'***
- Series of instructions/map created for physical devices (***'Code-a-Pillar' and 'BlueBots'***) or activities (Mazes)

Year Group: 1	Term: Spring 1 & 2	Topic: Computing Science	
Assessment			
Ongoing assessment from guided activities, observations, discussions, questioning and work evidence. A suggested activity is: • Code.org will track progress of children			
	x.1	**x.2**	**x.3**
Motion	Beginning to control motion by specifying the number of steps to travel, direction and turn.	Control motion by specifying the number of steps to travel, direction and turn.	Control motion by specifying the number of steps to travel, direction and turn with minimal or no debugging (corrections).
Looks	With adult support, can add text strings, show and hide objects and change the features of an object.	Add text strings, show and hide objects and change the features of an object.	Can add multiple text strings, show and hide objects and change the features of an object independently.
Sound	With adult support, can select sounds and control when they are heard, their duration and volume.	Select sounds and control when they are heard, their duration and volume.	Can independently select multiple sounds and control when they are heard, their duration and volume.
Draw	With adult support, can control when drawings appear and set the pen colour, size and shape.	Control when drawings appear and set the pen colour, size and shape.	Can independently control when drawings appear and set the pen colour, size and shape relevantly.
Events	With adult support, specify user inputs (such as clicks) to control events.	Specify user inputs (such as clicks) to control events.	Can independently specify multiple user inputs (such as clicks) to control events.

Control	With adult support, can specify the nature of events (such as a single event or a loop).	Specify the nature of events (such as a single event or a loop).	Can independently specify the nature of events (such as a single event or a loop).
Sensing	With adult support, can create conditions for actions by waiting for a user input (such as responses to questions like: What is your name?).	Create conditions for actions by waiting for a user input (such as responses to questions like: What is your name?).	Can independently create conditions for multiple and relevant actions by waiting for a user input (such as responses to questions like: What is your name?).