Hive Hackers - Section 1 - Algorithms - Lesson Plan

Section 1	Lesson 1 - Graph Paper Programming Lesson 2 - Real-life Algorithms Paper Planes	Lesson Duration	1 hour

Lesson Overview	During the lesson the students will be completing: - Lesson 1 - Graph Paper Programming - Lesson 2 - Real-life Algorithms Paper Planes on the code.org platform. The instructional video will walk you through this. Students are introduced to the concept of programming. Students will work along with the video to help create algorithms to shade squares. The goal is to reproduce an existing picture. The students will also create paper airplanes by working out the correct algorithmic sequence. If time permits, students will be able to decorate their plane.
Objectives	● Understand the difficulty of translating real problems into programs. ● Learn that ideas may seem clear, and yet still be misinterpreted by a computer. ● Practise communicating ideas through codes and symbols. ● Name various activities that make up their day. ● Decompose large activities into a series of smaller events. ● Arrange sequential events into their logical order.

Open either one of:

- Video Lesson 1 - Graph Paper Programming
- Powerpoint Lesson 1 - Graph Paper Programming

Starter (10 minutes)

Title	Description	Timeline
Introduction	Brief introduction to the facilitator and the course.	00:00 - 00:51
Video	Introducing ‘Computer Science’.	00:51 - 2:00
Introduction to lesson	Brief introduction on how the course will be run.	2:00 - 02:26
Vocabulary	Introduce the two words ‘Algorithm’ & ‘Program’ to students and if available, get them to record the definitions in their workbook.	02:26 - 03:50
Vocabulary Example - Alien Friend	An alien will give an example of following instructions to complete a task. Get the students to come up with their own algorithm for different tasks - Such as how to get ready for school or make a cake.	03:50 - 05:25

Main (20 minutes) - Lesson 1 - Graph Paper Programming

Title	Description	Timeline
Graph Paper Programming - Introduction	Ask the students how they would direct you around the square to fill in the black squares - To start they can use any vocabulary to give the simple steps.	05:25 - 06:25
Graph Paper Programming - Introduce Symbols	If available, get the students up and on their feet. Physically demonstrate the symbols with the whole class by getting students to action the instructions -	06:25 - 06:52

	Moving left, right, up, down and squiggle. For the squiggle action we can get the students to do a silly dance and shout “Squiggle!”	
Example 1 - Graph Paper Programming	Using these concepts and commands, demonstrating the example to the students. Explain how the squares are shaded in, by verbalising the algorithm aloud and turning the verbal instructions into a program of directions. Complete the examples and remember to highlight the ‘lazy programmer’ principle - That's the getting the fewest steps to complete the task! (Shown at 08:27 within the video). If available, distribute sheets of 4x4 graph paper or personal whiteboards. Ask the students to create an algorithm by shading-in black squares on pre-printed graph paper or on their personal whiteboards, based upon an image on screen. Discuss how each student may have different steps to complete the task and how there may be a longer and shorter way to get from A to B.	06:52 - 09:18
Example 2 - Graph Paper Programming	Get the students to complete example 2 using the same process as the previous example. If available, distribute sheets of 4x4 graph paper or personal whiteboards. Ask the students to create an algorithm by shading-in black squares on pre-printed graph paper or on their personal whiteboards, based upon an image on screen. Discuss how each student may have different steps to complete the task and how there may be a longer and shorter way to get from A to B. Using these concepts and commands, demonstrating the example to the students. Explain how the squares were shaded in, by verbalising the algorithm aloud and turning the verbal instructions into a program of directions.	09:18 - 11:10
Log into Code.org	As this is the first week ensure that the students can successfully log into the code.org platform. The instructions for this are located within the introduction slides! Complete Lesson 1 - Graph Paper Programming	11:10 - 11:30
Students complete the online task	**Allow Students to complete Lesson 1 - Graph Paper Programming. The answers for this are located within the Teacher Section of this course. Assist the students where possible.**	**15 minute independent working**

Main (30 minutes) - Lesson 2 - Real-life Algorithms: Paper Planes

Title	Description	Timeline
Introduction	Welcome the students to the second lesson. Recap of what we did in the last lesson - Including the key words ‘Algorithm’ & ‘Program’.	00:00 - 00:47
Code.org Introduction	Code.org will give an example of an algorithm - on how to make chocolate.	00:47 - 1:28
Examples	See if the students can come up with their own Algorithm and Program for a task - How about making a cup of tea or planting seeds to grow trees! Outline the different morning routine options, highlighting how there are many ways to complete the same task. Discuss with students how we are able to deconstruct a large task into a few logical steps.	1:28 - 2:10
Code.org Introduction	Code.org explain what we are doing in today's lesson - Following this we will walk the students.	2:10 - 03:10
The challenge Explained	Show the simple 2 steps to creating our program. Step 1 - remove what we don’t need. Step 2 - Order the remaining steps	03:10 - 03:54

The challenge - Solution for Step 1	Ask the students what steps are not needed and remove these.	03:54 - 04:30
The challenge - Solution for Step 2	Ask the students to put the 6 steps in the correct order.	04:30 - 05:17
Building your plane	Allow the students to create their paper airplane. Remember to follow the steps on screen as some students will create their very own.	05:17 - 05:45
Students build and design their paper airplane	**Allow Students to build their paper airplane.** Extension - Allow the students to design their paper airplane	**20 minute independent working**
Log into Code.org -	As this is the first week ensure that the students can successfully log into the code.org platform. The instructions for this are located within the introduction slides! Complete Lesson 2 - Real-life Algorithms Paper Planes	05:45 - 06:35
Plenary	Ask the students: - What they now know about our keywords - ‘Algorithm’ & ‘Program’? - Can they give an example of an algorithm or a program? - What was so important when we were creating our algorithm? Why did the alien get it wrong the first time Show the students what we will be doing next week	06:35 - 07:05

Extension Activities

Graph Paper Programming	If the workbook is available get the students to create algorithms for each of the tasks. The last few tasks the students can create their very own pattern and swap it with their peers. This provides an opportunity to be creative and encourage students to apply their knowledge. This activity could also be used as homework, allowing parents/guardians to get involved. Students could create their own image in class and as homework, then collaborate at home to produce the algorithm to code it.
Decorate Your Paper Plane	Some students can cut and stick the steps for the paper airplane algorithm into their book. Ask students to decorate their planes using the brightest colours possible. Encourage them to be creative as possible! This will provide enough time for all students to get to the same place.
Paper Airplanes - Create a paper plane using an algorithm:	Instruct students to cut-out the steps for making a paper airplane, as provided on the worksheet. Students should work together to choose the six correct steps from the total of nine options. Students will paste the correct steps, in order, onto the ‘paper airplane’ page in their workbook. Once students have completed this, they must take a blank sheet of white paper and construct their own paper airplane using the identified algorithm. Once all paper airplanes are made, take students in groups, or as a class outside to the playground to fly their paper airplanes.

Plenary (10 minutes)

Flash Chat: What did we learn?	● What if we used the same arrows, but replaced "Fill-In Square" with "Lay Brick"? ● What might we be able to do? ● What else could we program if we just changed what the arrows meant?
Feedback	Students give feedback on the lesson using the “two stars and a wish” framework
Reflection Sheet	Students should complete their reflection sheets for lesson 1.

Required Resources:

- Blank paper, markers, pens, or pencils

Optional Resources:

- Graph Paper Programming Activity Worksheet
- Scissors and glue
- Personal whiteboard/graph paper
- Real-Life Algorithms Worksheet: Paper Airplanes
- Student & Teacher Workbook for the lesson

Curriculum Links:

Desirable Features	**Interactive Design -** On paper, students will create algorithms for increasingly complex images. Students are encouraged to refine and develop these algorithms using the ‘lazy coder’ framework. Students will reflect upon the lesson in the worksheets provided in their booklets.
5 E’s link	**Evaluate –** Students will reflect on their progress and consider what went well in the ‘reflection sheet’. **Express –** Students are creative while performing the paper plane activity.
Possible Extensions	**Exchange -** Link with another school online via Google Hangouts. Use technology (a video conferencing platform such as Google Hangouts or Skype) to share student algorithms between schools. Each school should try and guess which image the algorithm is for. These algorithms could be for any of the six images provided in the student workbook. Create a video showing pupils flying their paper planes and exchange this video with another school. This could possibly be achieved using flipgrid.