DIGITAL SCHOOLHOUSE

+44 (0) 203 534 0580
info@digitalschoolhouse.org.uk

Enrichment Day Teaching Guide

Beautiful Numbers: Mathematical Art

Introduction

This workshop draws the link between Mathematics, Art and Computing and has been adapted from the workshop materials produced at Langley Grammar School. We look at the Fibonacci number sequence and the concept of the Golden Ratio and discuss how this has been applied to art and design throughout the ages, as well as being present in nature. Pupils are encouraged to explore the Fibonacci number sequence and how it is constructed. The use of mental maths is encouraged, especially in the morning to generate and test their number sequences as well as identify patterns within a given set of numbers.

Pupils are introduced to algorithms and asked to write out their own algorithm to calculate a Fibonacci number sequence. Before implementing their algorithm pupils are encouraged to dry run/test their code and participate in peer feedback. Once they have used logical reasoning to ensure their algorithm works, pupils will then be able to implement their algorithm.

This workshop makes use of Scratch as the programming environment; however, the skills and concepts delivered here can be implemented within any number of programming languages. One obvious alternative choice would be Python. In fact, for higher ability pupils, it is encouraged that the day is adapted to enable them to move onto Python quickly after testing the logic of their algorithm within Scratch.

ukie PlayStation SEGA Warwickshire County Council

Computing Programmes of Study Links

2.1 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

2.2 use sequence, selection, and repetition in programs; work with variables and various forms of input and output

2.3 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

2.6 select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

Progression Pathway bands covered

Write band abbreviation, full name followed by coloured levels/paths i.e.

P&D = Programming & Development: Pink, Yellow, Orange, Blue

Reference

PP2	Executes, checks and changes programs
PP3	Understands that programs execute by following precise instructions
YP1	Uses arithmetic operators, if statements, and loops, within programs.
YP2	Uses logical reasoning to predict the behaviour of programs
YP3	Detects and corrects simple semantic errors i.e. debugging, in programs.
OP1	Creates programs that implement algorithms to achieve given goals.
OP2	Declares and assigns variables.
OP3	Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.
BP1	Understands the difference between, and appropriately uses if and if, then and else statements.
BP2	Uses a variable and relational operators within a loop to govern termination.

ALG = Algorithms: Pink, Yellow, Orange

Reference

PA1	Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically.
PA2	Understands that computers need precise instructions.
PA3	Demonstrates care and precision to avoid errors
YA1	Understands that algorithms are implemented on digital devices as programs
YA2	Designs simple algorithms using loops, and selection i.e. if statements.
YA3	Uses logical reasoning to predict outcomes.
OA1	Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else.
OA2	Uses diagrams to express solutions.
OA3	Uses logical reasoning to predict outputs, showing an awareness of inputs.

IT = Information Technology: Pink, Yellow, Orange, Blue

Reference

PA1	Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically.
PA2	Understands that computers need precise instructions.
PA3	Demonstrates care and precision to avoid errors
YA1	Understands that algorithms are implemented on digital devices as programs
YA2	Designs simple algorithms using loops, and selection i.e. if statements.
YA3	Uses logical reasoning to predict outcomes.
OA1	Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else.
OA2	Uses diagrams to express solutions.
OA3	Uses logical reasoning to predict outputs, showing an awareness of inputs.

Computational Thinking Strands

AL – Algorithmic Thinking

Ref.	Activity
A1	Writing instructions that if followed in a given order (sequences) achieve a desired effect
A2	Writing instructions that use arithmetic and logical operations to achieve a desired effect
A3	Writing instructions that store, move and manipulate data to achieve a desired effect; (variables and assignment)
A4	Writing instructions that choose between different constituent instructions (selection) to achieve a desired effect;
A5	Writing instructions that repeat groups of constituent instructions (loops/iteration) to achieve a desired effect;
A7	Writing instructions that involve subroutines use copies of themselves to achieve a desired effect (recursion);

AB – Abstraction

Ref.	Activity
Ab1	Reducing complexity by removing unnecessary detail;
Ab2	Choosing a way to represent artefacts (whether objects, problems, processes or systems) to allow it to be manipulated in useful ways;

EV – Evaluation

Ref.	Activity
E1	Assessing that an algorithm is fit for purpose;
E2	Assessing whether an algorithm does the right thing (functional correctness);
E3	Designing and running test plans and interpreting the results (testing);
E8	Assessment of whether a system gives an appropriately positive experience when used (user experience);
E10	Stepping through algorithms/code step by step to work out what they do (dry run / tracing);

GE – Generalisation

Ref.	Activity
G1	Identifying patterns and commonalities in problems, processes, solutions, or data.

Learning Outcomes

1. To be able to understand what the Fibonacci number sequence is
2. To be able to mathematically create a Fibonacci number sequence
3. To understand the concept of the Golden Ratio and how it is derived from a division of two Fibonacci numbers
4. To understand the application of the Golden Ratio and the Fibonacci sequence within nature, design and architecture.
5. To be able to write an algorithm that calculates a Fibonacci number sequence
6. To understand what an algorithm is
7. To be able to represent an algorithm in a diagrammatical format such as a flowchart
8. To understand what is meant by the term variable
9. To be able to create and use variables within their program
10. To understand what is meant by the term 'selection' and 'loop'.
11. To be able to use selection and loop statements within their algorithm
12. To be able to use selection and loop statements within their program
13. To be able to dry run/test their algorithm to see if it works and achieves desired results
14. To be able to implement their algorithm as a program
15. To be able to understand recursion in its simplest form
16. To understand that we can implement the Fibonacci sequence using recursion in our programming
17. To be able to evaluate the effectiveness of their solution
18. To be able to use peer feedback to evaluate the effectiveness of their solution
19. To be able to identify areas for improvement and modification within their program
20. To be able to implement changes to their program, based on feedback and evaluation.

Session Overview

SESSION 1

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Welcome and Introduction	DSH_WelcomeIntroduction.ppt			
Starter Activity: "How to fake a super brain". Familiarise yourself with the mathemagic activity from CS4FN. However, in brief: Ask one pupil to write two numbers on the board (both under 30). Another then writes the sum of the two numbers below them. Find a system where the class contributes (or a single student) continues to add up the last two numbers and write them in a list of up to 10 numbers. During this time look away so they don't think you're doing the maths as you go along. Then finally ask another volunteer to mentally find the sum of all the numbers. While they are doing that, you beat them too it, by taking the fourth number from the bottom and multiplying it by 11. (add an extra zero and add 11 to the number!). Amaze them with your answer Challenge them to repeat it Alternatively, you could ask students to carry this out in pairs, or small groups after some key volunteers have been shown how to do it. Pull the class together, and explain the maths behind it. Although do see if they can work it out for themselves first	How to fake a super brain.doc Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA2, YA3	A1, G1	2.3
Explain the Fibonacci sequence in more detail. Describe its association with the Golden Ratio. Slides 4 to 13 will assist with this.	Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA2, YA3	A1, G1	2.3
Access the Template file, also available as an image	Template.docx	IT	A1, G1	2.6

Can they draw an image that fits around the Fibonacci sequence? Perhaps they could access the web cam to use their own self portrait. See how the sequence fits over it? Also available is a mask template – let students take their own picture and then using a photo editing software, to overlay the mask over their image. What can they find? It will inspire some strong discussion. This may be a good slot to use additional software if it's available. For example: "Fame Factors" app on iTunes http://www.intmath.com/numbers/math-of-beauty.php	Template.gif Mask-template.gif Beautiful Numbers.pptx	PI5, YI5 ALG PA3, YA3, OA2, OA3		
Explain the Fibonacci sequence and the mathematical algebra using slides 15 – 22	Beautiful Numbers.pptx	ALG PA1, YA3, OA2, OA3	A1, A2, A3, Ab1, Ab2	2.3
Discuss algorithms, remind the students the concept and work through slides 23 to 25. As an alternative you can ask students to write down a set of instructions for a routine activity that they do at home i.e. instructions for brushing their teeth. They should then swop the instructions with their partners and each should role play the activity. This will provide the pupil with instant visual feedback about the accuracy of their instructions and prompt them to instinctively spot and point out the errors (testing and debugging).	Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA2, YA3, OA1, OA2, OA3	A1, A2, A3, A4	2.1, 2.2, 2.3
Use slide 27 to help you explain variables. Ensure students understand this before you move on	Beautiful Numbers.pptx	P&D OP2, BP2	A1, A2, A3, A4	2.2
Use slide 28 to play word association. How many words can the students guess? Or alternatively, remove the animation to use it to help you describe the concepts to the pupils	Beautiful Numbers.pptx	P&D OP2, OP3, BP2	A1, A2, A3, A4	
Writing an algorithm for the Fibonacci sequence. Use slides 29 – 31 to help pupils use a flowchart to draw out their own algorithm for programming the Fibonacci sequence. If there is time before the break, as an extension activity ask the pupils to test out their algorithm by asking a peer to work through it with a sequence of numbers (dry run testing)	Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA2, YA3, OA1, OA2, OA3	A1, A2, A3, A4, E1, E2, E10	2.1, 2.2, 2.3

SESSION 2

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Recap key aspects from session 1. Describe how we are going to create a programme that calculates the Fibonacci sequence.	Beautiful Numbers.pptx			2.1, 2.2, 2.3
Introduce the concept of selection using slides 32 to 38. Enable pupils to build a simple program which executes and calculates the first Fibonacci number. Then build in the loop which enables multiple sequences. Use example files “FibonacciSeries” onwards.	FibonacciSeries.sb FibonacciSeries_N.sb Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA2, YA3, OA1, OA2, OA3 P&D PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP1, BP2	A1, A2, A3, A4, A5, Ab1, Ab2, G1	2.1, 2.2, 2.3
If the pupils are able to then briefly describe recursion. Use the YouTube video – show segments from it to enable the pupils to see the story. Slide 39 contains a button hyperlinked to the slide. Alternatively, the video has been downloaded from YouTube and is contained within this pack.	Dr Seuss – The Cat in a hat comes back.mp4 Beautiful Numbers.pptx	P&D PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP1, BP2	A7	2.2

SESSION 3

Session Content / Activity	Resources Used	Prog. Pathway	Comp. Thinking	Computing POS Link
Enable pupils to complete their solutions. Building in loops and variables.	Beautiful Numbers.pptx	P&D PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP1, BP2	A3	2.2
Pupils test their own solutions to identify if it works. Allow them time to correct their own errors.	Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA3, OA3 P&D PP2, PP3, YP2, YP3, OP2 IT PI5, YI5, OI3, BI4	E1, E2, E3, E8, E10	2.3
Enable pupils to group into 2's or 3's and test each other's solutions. Encourage them to provide feedback and assist with the debugging of it. Ask pupils to use and compare their solution with their original algorithm. This can assist with debugging, but they can also use it to highlight differences. Encourage them to discuss the differences in groups. Does the algorithm for each pupil differ? Does their solution work despite this?	Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA3, OA3 P&D PP2, PP3, YP2, YP3, OP2 IT PI5, YI5, OI3, BI4	E1, E2, E3, E8, E10	2.3, 2.6
Ask groups to consider and develop potential modifications to their solution. How could they make it increasingly: ➢ Interactive ➢ Animated ➢ Add additional functionality	Beautiful Numbers.pptx	ALG PA1, PA2, PA3, YA1, YA3, OA3 P&D PP2, PP3, YP1, YP2, YP3, OP1, OP2, OP3, BP1, BP2 IT PI5, YI5, OI3, BI4	G1, Ab1, Ab2	2.3, 2.6
End the session with pupils volunteering to demonstrate their solutions, explain what they did and what additional modifications they made and why	Beautiful Numbers.pptx		E1, E2, E3, E8, E10	2.3, 2.6

Files/Resources

Filename	Resource Type	Purpose/Description
1	Image	Source code image
2	Image	Source code image
5	Image	Source code image
8	Image	Source code image
13	Image	Source code image
Beautiful Numbers	PowerPoint	PowerPoint teaching resource
Dr. Seuss – Cat in a hat comes back	MP4 Video	Video resource to highlight recursion
Fib-basic	Scratch Source File	Source code – sample scratch project
FibonacciSeries	Image	Source code image
FibonacciSeries	Scratch Source File	Source code – sample scratch project
FibonacciSeries_N	Image	Source code image
FibonacciSeries_N	Scratch Source File	Source code – sample scratch project
FibonacciSeries_N_21	Image	Source code image
FibonacciSeries_N_21	Scratch Source File	Source code – sample scratch project
FibonacciSeries_N_list	Image	Source code image
FibonacciSeries_N_list	Scratch Source File	Source code – sample scratch project
FibonacciSeries_N_list-scratch	Image	Source code image
FibonacciSeries_N_Question	Image	Source code image
GoveExample	Scratch Source File	Source code – sample scratch project
How to fake a super brain	Word document, Information sheet	Information sheet for starter activity on slide 2
Mask-transparent	Image	Transparent overlay image
Maths-Basic	Scratch source file	Source code – sample scratch project
Template	Word document, worksheet	Fibonacci Spiral overlay template
Template	Image	Transparent overlay image

PLEASE NOTE: The activities outlined in this workshop pack are a suggested outline of how the workshop can be delivered. It is envisaged that teachers will adapt the resources and the organisation of them according to the needs of their class.