



Hour of Power

Students Will Be Able To

1. *Explain how computer science is shaping multiple industries.*
 2. *Experience basic computer science concepts.*
-

Materials

Video: [Computer Science is Changing Everything](#) [Run time is 5:34]

Varies, based upon activity.

Introduction

There's an old saying that "knowledge is power." In a world increasingly shaped by computer technology, it's important for students to gain a familiarity with basic computer programming concepts. The "Hour of Code"—celebrated the first full week of December—provides multiple opportunities for students to build this familiarity, but they can learn more any time by searching for "[Hour of Code activities.](#)" Today in Advisory, students can not only start to think about how computers are enhancing multiple industries, but can try out some basic computer science concepts.

Activity 1 – Introductory video

Begin by asking students which industries they're thinking of working in once they're done with their education. Have 5 or 6 students verbally share their ideas.

Screen the video [Computer Science is Changing Everything](#) . (Run time is 5:34). [I recommend that you show this video with the CC [closed caption] option, found on the lower right side of screen.]

Activity 2 – “20 Questions” Game

This game allows students to see that they already know how to search efficiently. It also introduces them to the basic concept of “divide & conquer” problem solving by using a “search algorithm.” [An “algorithm” is a set of instructions that you give to a computer to complete a





task.] Students can also think about the idea of “efficiency analysis” as a way to compare algorithms. Which is more efficient?

Play a game of “20 Questions.” The person leading the game thinks of a famous person and the class guesses their identity. **They can only use “yes/no” questions, and they can only use 20 questions total.** As the class plays, encourage them to think about what makes a good question.

Using a British class as an example, the instructor pointed out that the class didn’t ask questions like: “Is it Adele?” or “Is it Usain Bolt?” until quite late. Why not?

What is a great first question? What makes it so good?

A good question will rule out half of the possibilities, whatever answer you receive. To do well at playing “20 Questions,” you need to ask questions that rule out half of the possibilities each time.

If you ask single-elimination questions each time (which is what “linear searching” is), it takes quite a while to get to the correct answer. If you ask questions that help you to eliminate half of the possibilities each time (also known as binary searching), the process goes much faster. Each time, you cut in half the amount of data needed to solve the problem. Utilizing the process of divide & conquer problem solving [with a “divide & conquer” algorithm], you can be much more efficient.

[Source: <https://teachinglondoncomputing.files.wordpress.com/2014/01/activity-20questions.pdf>]

Activity 3 – “Loops”

“Loops” are a way to be more efficient while programming computers.

This video explains how this concept works:

[“Unplugged Activity: Getting Loopy”](#) (Run time is 2:17)

Students can work out a basic line dance, and then capture it on the board using the “loop” concept as a way to efficiently repeat certain actions.

Activity 4 – Snowflake Doodle Art

Materials needed: a **blank piece of paper** and a **writing utensil** for **each** student.

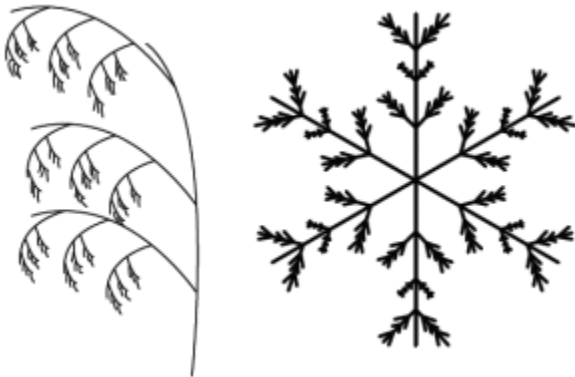
Concepts include algorithmic thinking and recursion.





ALGORITHMIC DOODLE ART

Follow simple algorithms involving fractions to draw pictures reminiscent of nature.



Scenery in films is often computer generated. Ever wondered how they do it? Next time you find yourself drawing doodles, draw an algorithmic doodle and explore algorithms for drawing nature. The algorithms are recursive: that is they describe one step and then tell you just to draw the next step in the same way, following the algorithm from the start.

Example

Here is an example of a Doodle Algorithm. What does it look like when you're done?

1. Start by drawing a single straight vertical line
2. Doodle draw from that line as follows.

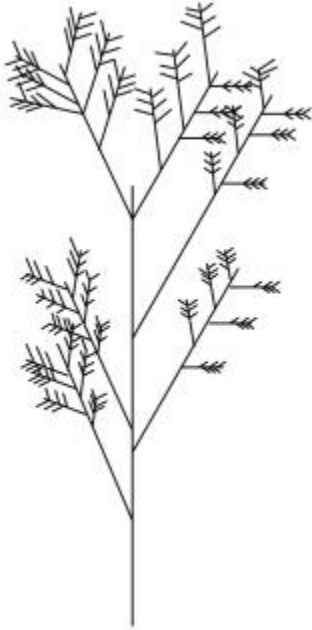
To doodle draw from a given line:

1. Draw 3 shorter lines at an angle in the **top two-thirds** of the line on its **left** side.
2. Draw 3 shorter lines at an angle in the **top two-thirds** of the line on its **right** side.
3. Choose a new existing line and **doodle draw** from that line.





By following the algorithm you should get a picture something like this one, though as you choose the exact positions and lengths of lines randomly yours won't be exactly the same just as no two plants are identical.



Different algorithms give different looking trees, grasses, ferns, snowflakes, crystals...

[Source: <https://teachinglondoncomputing.org/algorithmic-doodle-art/>]

