Suggested Coding Progression for Schools K-7

K-2: Computer Science Fundamentals: Unplugged Programming

Applied Design Skills and Technology:

Big Ideas:

Designs grow out of natural curiosity.	Skills can be developed through play.	Technologies are tools that extend human capabilities.

Curricular Content:

Making

· Choose tools and materials

· Use trial and error to make changes, solve problems, or incorporate new ideas from self or others

Sharing

· Demonstrate their product, tell the story of designing and making their product, and explain how their product contributes to the individual, family, community, and/or environment

· Reflect on their ability to work effectively both as individuals and collaboratively in a group

Applied Skills

· Use materials, tools, and technologies in a safe manner in both physical and digital environments

· Develop their skills and add new ones through play and collaborative work

Applied Technologies

- Explore the use of simple, available tools and **technologies** to extend their capabilities

Unplugged Programming teaches students computer science fundamentals without a computer. Fundamentals can be taught to any grade level, either as a self-contained unit or before moving on to programming on computers or tablets. Short assessments are available for each lesson. Each lesson correlates with aspects of the Applied Design, Skills, and Technology curriculum for K-1.

Lesson 1: Happy Maps

Students create a set of instructions to move a character through a maze. Scissors and glue required!

Objectives:

- List steps to move character around a map.
- Arrange directions to reach predetermined goal.
- Predict where character will land, given a list of steps.

Lesson 2: Move It, Move It

Students learn the cardinal directions and do a paper based activity. They'll need glue and scissors. Students use hand gestures to communicate and "program" each other to move around a space. Best done in the gym!
Objectives:
Recognize situations where they can create programs to complete tasks.

- Predict moves necessary to get teammate from start to finish.
- Convert movements into symbolic instructions.
- Relate algorithms as programs to teammate.

Lesson 3: Brown Bear, Brown Bear Algorithms
Using the story Brown Bear, Brown Bear, What Do You See?, students create an algorithm using arrows to follow the order of the story.

Lesson 4: Real Life Algorithms
Students relate programming to everyday activities like planting a seed. This is a put the steps in order activity. Scissors and glue needed!

Lesson 5: Crazy Character Algorithms
Students follow a set of teacher given instructions to draw a crazy character using different shapes. Each student will draw something different; the important thing for students to understand how important it is to be specific when giving instructions.

Important vocabulary at this stage:
Algorithm: a set of instructions
Command: a specific instruction given to a computer to perform a task
Program: an algorithm that has been coded into something that can be run by a machine
Bug: a mistake or problem in your algorithm
Debug: fixing a mistake or problem in your algorithm

This work can also be supplemented by short activities on the iPad that are visual, no text, perfect for early learners!

Grade 3-5: Computer Science Fundamentals: Unplugged Programming

Grade 3 Applied Design Skills and Technology:
Big Ideas:

Designs grow out of natural curiosity.	Skills can be developed through play.	Technologies are tools that extend human capabilities.

Curricular Content:
Making

- Choose tools and materials

· Use trial and error to make changes, solve problems, or incorporate new ideas from self or others

Sharing

· Demonstrate their product, tell the story of designing and making their product, and explain how their product contributes to the individual, family, community, and/or environment

· Reflect on their ability to work effectively both as individuals and collaboratively in a group

Applied Skills

· Use materials, tools, and technologies in a safe manner in both physical and digital environments

· Develop their skills and add new ones through play and collaborative work

Applied Technologies

- Explore the use of simple, available tools and **technologies** to extend their capabilities

Grade 4-5 Applied Design, Skills, and Technology

Big Ideas:

Skills are developed through practice, effort, and action.		The choice of technology and tools depends on the task.

Curricular Competencies

Defining

- Identify key features or user requirements
- Identify the main objective for the design and any **constraints**

Ideating

- Generate potential ideas and add to others' ideas
- Screen ideas against the objective and constraints
- Choose an idea to pursue

Prototyping

- Outline a general plan, identifying tools and materials
- Construct a first version of the **product**, making changes to tools, materials, and procedures as needed
- Record **iterations** of prototyping

Testing

- Test the product
- Gather peer feedback and inspiration

Lesson 1: Graph Paper Programming

By "programming" one another to draw pictures, students will begin to understand what programming is really about. The class will begin by having students instruct each other to color squares in on graph paper in an effort to reproduce an existing picture. If there's time, the lesson can conclude with images that the students create themselves.

Mini Lessons 1-3 (for younger classes)

1) Giving grids and following directions

2) Creating directions and writing for friend

3) Assessment

Lesson 2: Group Grid Programming

In groups of 4, students use place markers to navigate their token into a designated home square, using basic commands (arrows learned from graph paper programming)

Lesson 3: Lego Coding

Students work in small groups to create a maze made of Lego and then navigate their Lego character through the maze using basic commands (move forward, turn left, turn right)

Lesson 4: Relay Programming

Big Idea (Gr. 3) : Having good communication skills and managing our emotions enables us to develop and maintain healthy relationships.

Big Idea (Gr 4/5): Daily physical activity enables us to practice skillful movement and helps us develop personal fitness.

In a large room (gym), students will quickly move to a race against the clock and work together to create a program, one instruction at a time. Review of Graph Paper Programming

Lesson 5: Getting Loopy

Loops are a handy way of describing actions that repeat a certain number of times. Students are introduced to loops by practicing converting sets of actions into a single loop.

Lesson 6: Conditionals with Cards

We don't always know ahead of time what things will be like when we run our computer programs. Different users have different needs, and sometimes you will want to do something based off of one user's need that you don't want to do with someone else. That is where conditionals come in. This lesson demonstrates how conditionals can be used to tailor a program to specific information.

Grade 5-7: Computer Science Fundamentals: Unplugged Programming

Grade 4-5 Applied Design, Skills, and Technology

Big Ideas:

Skills are developed through practice, effort, and action.		The choice of technology and tools depends on the task.

Curricular Competencies

Defining

- Identify key features or user requirements
- Identify the main objective for the design and any **constraints**

Ideating

- Generate potential ideas and add to others' ideas
- Screen ideas against the objective and constraints
- Choose an idea to pursue

Prototyping

- Outline a general plan, identifying tools and materials

- Construct a first version of the **product**, making changes to tools, materials, and procedures as needed
- Record **iterations** of prototyping

Testing

- Test the product
- Gather peer feedback and inspiration

Grade 6-7 Applied Design, Skills, and Technology

Design can be responsive to identified needs.	Complex tasks require the acquisition of additional skills.	Complex tasks may require multiple tools and technologies.

Curricular Competencies

Testing

- Gather peer and/or user and/or expert feedback and inspiration
- Make changes, troubleshoot, and test again

Making

- Identify and use appropriate tools, **technologies**, and materials for production
- Make a plan for production that includes key stages, and carry it out, making changes as needed
- Use materials in ways that minimize waste

Sharing

- Decide on how and with whom to **share** their product
- Demonstrate their product and describe their process, using appropriate terminology and providing reasons for their selected solution and modifications
- Evaluate their product against their criteria and explain how it contributes to the individual, family, community, and/or environment
- Reflect on their design thinking and processes, and evaluate their ability to work effectively both as individuals and collaboratively in a group, including their ability to share and maintain an efficient co-operative work space
- Identify new design issues

Applied Skills

- Identify and evaluate the skills and skill levels needed, individually or as a group, in relation to a specific task, and develop them as needed

Lesson 1: Graph Paper Programming

By "programming" one another to draw pictures, students will begin to understand what programming is really about. The class will begin by having students instruct each other to color squares in on graph paper in an effort to reproduce an existing picture. If there's time, the lesson can conclude with images that the students create themselves.

Lesson 2: Ping Pong Rescue

Big Ideas: - Design can be responsive to identified needs

- *Complex tasks require the acquisition of additional skills*
- *Complex tasks may require multiple tools and technologies*

Core Competencies

Communication

- *Connect and engage with others (to share and develop ideas)*
- *Collaborate to plan, carry out, and review constructions and activities*
- *Explain/recount and reflect on experiences and accomplishments*

Critical Thinking

- *Analyze and critique*

Curricular Competencies

Ideating

- *Generate potential ideas and add to others' ideas*

Content

Computational Thinking

- *Simple algorithms that reflect computational thinking*

The ability to sequence instructions into simple algorithms is fundamental to computational thinking. In this activity, students work in teams of 2 to 4 to create a simple algorithm that guides one of their team members, who is blindfolded, to rescue a ping-pong ball. At the same time, students develop their core competencies in communication and critical thinking, as well as their curricular competency in ideating, which involves generating potential ideas or adding to others' ideas, screening ideas against criteria or constraints, and choosing an idea to pursue.

Lesson 3: Binary Numbers

The binary number system plays a central role in how information of all kinds is stored on computers. Understanding binary can lift a lot of the mystery from computers, because at a fundamental level they're really just machines for flipping binary digits on and off. There are several activities on binary numbers in this document, all simple enough that they can be used to teach the binary system to anyone who can count! Generally children learn the binary system very quickly using this approach, but we find that many adults are also excited when they finally understand what bits and bytes really are.

Lesson 4: Image Representation

Images are everywhere on computers. Some are obvious, like photos on web pages and icons on buttons, but others are more subtle: a font is really a collection of images of characters, and a fax machines is really a computer that is good at scanning and printing.

This activity explores how images are displayed, based on the pixel as a building block. In particular, the great quantity of data in an image means that we need to use compression to be able to store and transmit it efficiently. The compression method used in this activity is based on the one used in fax machines, for black and white images.

Lesson 5: Information Theory

Computers are all about storing and moving information, but what actually is information? How do we measure the amount of information in a message?

This activity uses some intriguing variations on the game of 20 questions to demonstrate how we can quantify information content, which in turn shows us how to store and share it efficiently.