SD#59 ADST - Computational Thinking Scope & Sequence

Computational Thinking (CT) is the ability to solve a problem by reducing it down to discrete steps that a machine or computer can solve. Students who learn CT can apply these skills across all subjects as well as life outside of the classroom. Coding is just one aspect of this.

Article: What is the difference between coding and computational thinking

ADST Scope and Sequence (K-3)

Big Ideas:

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

Curriculum Competencies:

Applied Design

Ideating

The goals are about taking an idea based on experience or interest, and exploring what is necessary to express and design that idea. Ideas from others can and should also be explored.

Making

In order to make a product that expresses an idea, tools and materials must be chosen. As well, an emphasis exists on iterating and changing through trial and error.

Sharing

Sharing doesn't just mean social media. It means being able to tell the story of your idea, and deciding not only the best medium, but also the best venue for sharing. Group work is encouraged.

Applied Skills

- Use materials, tools, and technologies in a safe manner in both physical and digital environments
- Develop their skills and new ones through play and collaboration

Applied Technologies (see available resources section)

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

Content & Computational Thinking

Students are expected to use the learning standards for Curricular Competencies from Applied Design, Skills, and Technologies K-3 in combination with grade-level content from other areas of learning in cross-curricular activities to develop foundational mindsets and skills in design thinking and making.

Available Resources at the DRC

Blue-Bot Robots
Code & Go Robot Mouse
Littlecodr Card Game
Creating with Kibo   website
Let's Go Code! Activity set

Various coding books for guided reading and teacher resource books included in the kits or to sign out separately

Maker Kits and Resources

iPad Apps:

- Blue-Bot
- Lightbot Jr./Lightbot
- Block Island
- Scratch Jr.

Online:

- Code.org (Pre-Reader Course A/B)
- Hour of Code

Books:

- Hello Ruby   website
- No Fear Coding: Computational Thinking Across K-5

Project Mapping Template

SD#59 ADST - Computational Thinking Scope & Sequence

Computational Thinking (CT) is the ability to solve a problem by reducing it down to discrete steps that a machine or computer can solve. Students who learn CT can apply these skills across all subjects as well as life outside of the classroom. Coding is just one aspect of this.

Article: What is the difference between coding and computational thinking

ADST Scope and Sequence (Grade 4-5)

Big Ideas:

- Designs can be improved with prototyping and testing.
- Skills are developed through practice, effort, and action.
- The choice of technology and tools depends on the task.

Curriculum Competencies:

Applied Design

Context Defining
- Choosing a design opportunity
- Identify key features or user requirements
- Identify 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(gathering peer feedback and inspiration)
- Make changes and test again, repeating until satisfied with the product

Making
- Construct the final product, with planned changes

Sharing
- Decide on how and with whom to share their product
- Demonstrate product and describe process
- Determine if product meets the objective and contributes to the individual, family, community, and/or environment
- Reflect on design thinking and processes
- Work effectively individually and in a group
- Share and maintain a co-operative work space
- Identify new design issues

Applied Skills:
- Use materials, tools, and technologies in a safe manner, and with an awareness of the safety of others
- Identify and develop the skills required for a task

Applied Technologies:(see available resources section)
- Use familiar tools and technologies to extend their capabilities when completing a task
- Choose appropriate technologies for specific tasks
- Show a willingness to learn new technologies as needed

Content & Computational Thinking

Students are expected to use the learning standards for Curricular Competencies from Applied Design, Skills, and Technologies 4-5 in combination with grade-level content from other areas of learning in cross-curricular activities to develop foundational mindsets and skills in design thinking and making.

Available Resources at the DRC

Snap Circuits
littleBits website
Spheros website
Edison Robots website
Micro:bits website
MakeyMakey website
Coding with Dash and Dot website
Lego WeDo 2.0 website
Scratch

Maker Kits and Resources

Unplugged:
- Robot Turtle Board Game website
- Code Master Board Game website

iPad Apps:
- Lego Mindstorms Fix the Factory
- Box Island
- Code Monkey
- Lightbot Hour
- Swift

Books:
- The Official Scratch Jr. Book
- Coding Games in Scratch
- Coding Projects in Scratch website
- Kids, Code & Computer Science Professional Journal

Project Mapping Guide

SD#59 ADST - Computational Thinking Scope & Sequence

Computational Thinking (CT) is the ability to solve a problem by reducing it down to discrete steps that a machine or computer can solve. Students who learn CT can apply these skills across all subjects as well as life outside of the classroom. Coding is just one aspect of this.

Article: What is the difference between coding and computational thinking

Coding Scope and Sequence (Grade 6-7)

Big Ideas:

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

Curricular Competencies:

Applied Design

Understanding Context
- Empathize with potential users to find issues and uncover needs and potential design opportunities

Defining
- Choose a design opportunity
- Identify key features or potential users and their requirements
- Identify criteria for success and any constraints

Ideating
- Generate potential ideas and add to others' ideas
- Screen ideas against criteria and constraints
- Evaluate personal, social, and environmental impacts and ethical considerations
- Choose an idea to pursue

Prototyping
- Identify and use sources of information
- Develop a plan that identifies key stages and resources
- Explore and test a variety of materials for effective use
- Construct a first version of the product or a prototype, as appropriate, make changes to tools, materials, and procedures as needed
- Record iterations of prototyping

Testing
- Test the first version of the product or the prototype
- 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
- 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
- Evaluate ability to work effectively individually and in a group and share and maintain a co-operative work space
- Identify new design issues

Content & Computational Thinking

Computational Thinking:
-Simple algorithms that reflect computational thinking
-Visual representations of problems and data
-Visual programming

Coding [Grade 6]:
-Understanding simple algorithms
-Be able to understand the concepts of inputs and outputs

Coding [Grade 7]:
-Draw and use a flow chart
-Use if, then statements
-Understand expressions "AND" "OR" "NOT"

Available Resources at the DRC

Snap Circuits
littleBits website
Spheros website
Edison Robots website
Micro:bits website
MakeyMakey website
Coding with Dash and Dot website
Lego Mindstorms EV3 website
Scratch
Maker Kits and Resources

Unplugged:
- Robot Turtle Board Game website
- Code Master Board Game website
- Littlecodr Cards
- Scratch Coding Cards

iPad Apps:
- Lego Mindstorms Fix the Factory
- Box Island
- Code Monkey
- Lightbot Hour
- Swift

Books:
- 20 Makey Makey Projects
- Coding for Beginners Using Scratch
- Coding Projects in Scratch website
- Getting Started with Raspberry Pi
- Kids, Code & Computer Science Professional Journal

Project Mapping Guide

SD#59 ADST - Computational Thinking Scope & Sequence

Computational Thinking (CT) is the ability to solve a problem by reducing it down to discrete steps that a machine or computer can solve. Students who learn CT can apply these skills across all subjects as well as life outside of the classroom. Coding is just one aspect of this.

Article: What is the difference between coding and computational thinking

Coding Scope and Sequence (Grade 8/9)

Big Ideas:

- Design can be responsive to identified needs.
- Social, ethical, and sustainability considerations impact design.
- Complex tasks require the acquisition of additional skills.
- Complex tasks may require multiple tools and technologies.

Curricular Competencies:

Applied Design

Defining
- Choosing a design opportunity
- Identify potential users and relevant contextual factors
- Identify criteria for success, intended impact and constraints

Ideating
- Take creative risks in generating ideas and add others ideas in ways that enhance them
- Screen ideas against criteria and constraints
- Critically analyse prioritize competing factors, including social, ethical, and sustainability considerations, to meet community needs for preferred futures
- Choose an idea to pursue, keeping other potentially viable ideas open

Prototyping
- Identify and use sources of inspiration and information
- Choose a form for prototyping and develop a plan that includes key stages and resources
- Prototype, making changes to tools, materials, and procedures as needed
- Record iterations of prototyping

Testing
- Identify sources of feedback
- Develop an appropriate test of the prototype
- Conduct the test, collect and compile data, evaluate data, and decide on changes
- Iterate the prototype or abandon the design idea

Content & Computational Thinking

Create algorithms that can be reliably repeated by others

Debug algorithms and programs by breaking problems down into a series of smaller problems

Identify different programming languages including:
- Visual programming (e.g. scratch)
- Text-based programming (e.g. html)
- Programming modular components e.g.(lego mindstorms)

Use binary code to digitally represent various data types including: text, sound, pictures, video

Develop and collaborate in a cloud-based environment

Use “ IF, THEN, ELSE” Statements within programs

Available Resources at the DRC

Raspberry Pi website
Arduino website
Lego Mindstorms EV3 Kit   website
Khan Academy HTML programing course
Wolfram programing lab:
http://www.wolfram.com/programming-lab

Books:
- Everyone Can Code Level 1 and 2
https://images.apple.com/education/docs/Get_Started_with_Code_Curriculum_Guide.pdf
- Raspberry Pi Project Workbook
- Adventures in Raspberry Pi

SD#59 ADST - Computational Thinking Scope & Sequence

Computational Thinking (CT) is the ability to solve a problem by reducing it down to discrete steps that a machine or computer can solve. Students who learn CT can apply these skills across all subjects as well as life outside of the classroom. Coding is just one aspect of this.

Article: What is the difference between coding and computational thinking

Coding Scope and Sequence (Grade 10)	Content & Computational Thinking
Big Ideas: ● Social, ethical, and sustainability considerations impact design. ● Complete tasks require the sequencing of skills ● Complex tasks require different technologies and tools at different stages. **Curricular Competencies:** Students are expected to do the following: **Applied Design** Engage in a period of **research** and **empathetic observation** in order to understand design opportunities Defining - Choose a design opportunity - Identify potential users and relevant contextual factors - Identify criteria for success, intended impact, and any constraints Ideating - Take creative risks in generating ideas and add to others' ideas in ways that enhance them - Screen ideas against criteria and constraints - Critically analyze and prioritize competing factors, including social, ethical, and sustainability considerations, to meet community needs for preferred futures - Choose an idea to pursue, keeping other potentially viable ideas open Prototyping - Identify and use sources of inspiration and information - Choose a form for prototyping and develop a plan that includes key stages and resources - Evaluate a variety of materials for effective use and potential for reuse, recycling, and biodegradability - Prototype, making changes to tools, materials, and procedures as needed - Record iterations of prototyping	Use more than one input to generate a suitable output. Using algorithms and/or commands to create programs that can solve common problems Apply research to applications to add flexibility and customizability to programs
	Available Resources
	Beginning/ Review Resources: Code.org (Puzzle block coding) Codecademy.com (Javascript coding) Other Recommendations: Legacy programming languages (Fortran, Cobol), i) teaching (e.g. Pascal, Modula) ii) object-oriented (e.g. C++, JAVA) iii) scripting (e.g. PERL, CGI) iv) meta-languages (e.g. HTML, JavaScript) Raspberry Pi website Arduino website Wolfram programming lab: http://www.wolfram.com/programming-lab

SD#59 ADST - Computational Thinking Scope & Sequence

Computational Thinking (CT) is the ability to solve a problem by reducing it down to discrete steps that a machine or computer can solve. Students who learn CT can apply these skills across all subjects as well as life outside of the classroom. Coding is just one aspect of this.

Article: What is the difference between coding and computational thinking

Coding Scope and Sequence (Grade 11 and 12)	Content & Computational Thinking
Big Ideas ● Products can be designed for life cycle. ● Personal design interests require the evaluation and refinement of skills. ● Tools and technologies can be adapted for specific purposes. **Curricular Competencies** Students are expected to be able to do the following: **Applied Design** *Understanding context* - Conduct **user-centred research** to understand design opportunities and barriers Defining - Choose a design opportunity and point of view - Identify potential users, intended impact, and possible unintended negative consequences - Make inferences about premises and boundaries that define the design space Ideating - Take creative risks to identify gaps to explore as design space - Generate ideas to create a range of possibilities and add to others' ideas in ways that create additional possibilities - Critically analyze how competing social, ethical, and sustainability considerations impact designed solutions to meet global needs for preferred futures - Prioritize ideas for prototyping and designing with users Prototyping - Identify and use a variety of sources of inspiration and information - Choose an appropriate form, scale, and level of detail for prototyping, and plan procedures for prototyping multiple ideas - Analyze the design for life cycle - Construct prototypes, making changes to tools, materials, and procedures as needed - Record iterations of prototyping	Simplify programming language to be concise for faster processing speeds (e.g. Repeat Loops, For Loops, While Loops) Integration between programming platforms (e.g. integrating a Javascript function and file into an HTML webpage) Debug Independently: the ability to identify errors in syntax and correct mistakes Design programs for real life scenarios
	Available Resources
	Beginning/ Review Resources: Code.org (Puzzle block coding) Codecademy.com (Javascript coding) Other Recommendations: Legacy programming languages (Fortran, Cobol), i) teaching (e.g. Pascal, Modula) ii) object-oriented (e.g. C++, JAVA) iii) scripting (e.g. PERL, CGI) iv) meta-languages (e.g. HTML, JavaScript) Raspberry Pi website Arduino website