Teaching for the Digital Future:

Developing a Pan-Canadian K-12 Computer Science Framework

Working Document 2: Draft Competence Guide

For Consultation

DIGITAL SKILLS FIT FOR OUR TIME

Technology underpins many of the tools and services we rely on every day, shaping the ways we work, communicate, shop, and play.

And though the best digital technologies are almost effortless to use, few of us understand how they work, or the logic that underpins them.

As digital technologies become increasingly embedded in our lives, future generations will need to be equipped with technological skills.

Canadian students need access to high-quality Computer Science education that will empower them to be both users and makers of digital technologies and enable them, with their abilities, to enjoy and harness the power of these new tools.

DEVELOPING A COMPUTER SCIENCE EDUCATION FRAMEWORK

In 2019, Canada Learning Code, working with a team of advisors, launched Teaching for the Digital Future: A Pan-Canadian Framework for K-12 Computer Science Education.

In particular, the Framework would:

- Be built by engaging with people from a variety of backgrounds;
- Present a vision for Computer Science education in Canada; and
- Provide a set of guidelines for what every Canadian student must know in order to navigate an increasingly digital world.

To that end, in summer 2019, we published our first Working Document, which introduced the project and outlined its vision for Computer Science education in Canada.

Following that, we launched a series of engagement activities to collect feedback and to seek advice on how to make the Framework most effective and identify what all Canadian students should know about Computer Science.

THE PROCESS OF DEVELOPING THE FRAMEWORK

Working with an Advisory Group of leading thinkers from across the country, Canada Learning Code is spearheading the development of this Framework by engaging with policy-makers, teachers, industry leaders, and NGOs.

2018 Forum

In November, we convened experts in Computer Science, as well as in policy and curriculum development, to discuss the need for a CS Framework, what it could include, and how it should be built.

Fall 2018

1

Research and Informant Interviews

We completed a scan of provincial and territorial curricula, researched similar frameworks from other jurisdictions, convened an advisory group, and spoke to some 30 key informants from Canada and around the world to gather recommendations on how to develop the Framework.

Winter/Spring 2019

Working Document 1: Initial Findings and Directions

2

Survey and Stakeholder Engagement

We want to make sure the perspectives of policy-makers, curriculum developers, teachers' unions, teachers, industry experts, and NGOs are captured. That's why we're gathering feedback on this working document through an online survey and workshops.

Summer/Fall 2019

Working Document 2: Competence Guide

3

Survey and Public Engagement

We will continue gathering feedback as we develop the Framework. After we release our second working document, we will continue collecting feedback through another online survey and workshops.

Winter/Spring 2020

Framework

WORKING DOCUMENT 2: OUR DRAFT COMPETENCE GUIDE

This document builds on our first Working Document and presents a draft competence guide. Ultimately, it will allow us to continue to collect feedback as we refine our ideas and develop the Framework.

This document is intended to:

- Articulate the five main components that form a comprehensive approach to Computer Science education;
- Propose 23 core Computer Science skills and competencies that students should demonstrate by the time they graduate high school;
- Describe some possible learning trajectories to clarify how a beginner, novice, and intermediate learner might be introduced and levelled up to the full skill or competency; and
- Support conversations with educators, policy-makers, Computer Science experts, and others, and launch a public survey to collect feedback, input, and advice about the skills and competencies all Canadian students should develop.

OUR METHODS FOR DEVELOPING WORKING DOCUMENT 2

Reviewed existing Computer Science curricula across Canada

Reviewed five existing and respected international Computer Science Frameworks

Integrated key learnings from workshops with policy-makers, industry experts, and teachers

Components of Computer Science education

THE FIVE MAIN COMPONENTS OF OUR COMPUTER SCIENCE FRAMEWORK

Computer Science education is more than just coding.

A comprehensive approach to Computer Science education includes learning skills and competencies from each of the following areas:

Coding and Programming

- Algorithms
- Data Structure
- Modelling & Abstraction
- Modularity
- Debugging

Computing and Networks

- Hardware & Software
- File Management
- Troubleshooting
- Digital Connectivity
- Cybersecurity

Data

- Data & its Uses
- Organizing Data
- Assessing Information
- Ownership & Governance
- AI & Machine Learning

Technology and Society

- Safe Communication
- Ethics, Safety, & the Law
- Tech & the Environment
- History & Technology
- Tech & Wellbeing

Design

- User Design
- Visual Design
- Universal Design

Draft Competence Guide

CAPACITIES AND DISPOSITIONS

In addition to teaching important skills and competencies, Computer Science education will also equip students with the capacities and dispositions required to meet the needs of their times.

Discovery: Computer Science education will inspire students to approach problems with curiosity and a sense of discovery. It will encourage students to try new things, approach tasks with a growth mindset, and iterate as they master new skills.

Critical Thinking: Having a better understanding of how computers operate will equip students to more critically engage the social, legal, ethical, and political implications of technology.

Team Work: Computer Science encourages students to work together and develop projects, promoting effective communication, project management skills, and empathy for the perspective of others. It also encourages students to constructively give and receive feedback and fosters a willingness to help and share with others.

Citizenship: Computer Science education will help students understand the ways in which technology impacts society, helping them to become technological stewards who will harness the power of technology to improve the world around them.

Resilience: Creating digital artifacts will help students to become more comfortable with trying new things, making mistakes, and learning through experience. By stressing the importance of continuous and unexpected learning, students will ultimately become more resilient and see opportunity in failure.

Creativity: Computer Science will encourage students to explore their creativity, think outside the box, and develop innovative solutions to address issues that affect them, their community, and the world.

HOW TO READ THE COMPETENCE GUIDE

Focus Area
This is one of the five major components of Computer Science education.

Grade 12 Achievement
This refers to what students should be able to achieve by grade 12. This achievement will demonstrate that students possess core Computer Science skills and competencies.

Skills and competencies all students should develop
These are the skills and competencies that we're proposing all students should master by grade 12.

Learning Progressions
These provide suggestions for how a student might progress through a particular core concept in order to fully acquire the skill and competency.

Additional Skills and Competencies
These are suggestions for the types of skills and competencies that students interested in pursuing higher education, a professional degree, or a job in Computer Science should also consider mastering

CODING AND PROGRAMMING

By Grade 12 students should be able to write a simple computer program using a text-based language and re-using existing code or remixing other programs.

To do this, students will need to possess the following skills and competencies:

	Algorithms	Data Structures	Modularity	Modelling & Abstraction	Debugging
Skill Unlocked	**Create an algorithm using a text-based language that includes functions, objects, conditionals, and arrays.**	**Build a data structure that can be used in an algorithm.**	**Create a module using a text-based language that can be applied in multiple programming contexts.**	**Evaluate existing models by identifying possible biases and using new data.**	**Explain how to test software for bugs and find solutions to problems they can anticipate.**
Almost There	Use loops, condition statements, and functions to create and reconstruct existing algorithms and improve their efficiency.	Explain how complex data structures (i.e., lists and maps) work and are used in algorithms. Describe their strengths and weaknesses.	Create a simple program using modules that already exist.	Create a model by observing, organizing, and analyzing patterns.	Analyze their own code and provide solutions to errors they've identified.
Keep Going	Create a simple algorithm to produce a certain action and describe other algorithms that can achieve the same result.	Explain how data structures (i.e., numbers and strings) work and how they are used to create algorithms. Describe their strengths and weaknesses.	Create modules by organizing a list of instructions into clusters using a visual programming language.	Identify relevant and irrelevant characteristics of objects and categorize them by their properties.	Solve errors in teacher-given text-based code.
Start Here	Provide the definition of an algorithm and explain its different components: loops, objects, condition statements, and functions.	Provide the definition of a data structure and its different functions: organizing, processing, retrieving, and storing data.	Recognize ways in which a large task can be broken down into a series of sub-tasks (i.e., modules).	Recognize patterns in the world and define common attributes of similar objects.	Debug programs built with a visual programming language.

For Bonus Points:

Document and justify computational processes when creating a program.

Describe the software development lifecycle.

Create more complex computer programs using more sophisticated algorithms and data structures.

Employ principles of writing maintainable code.

CODING AND PROGRAMMING

By Grade 12 students should be able to write a simple computer program using a text-based language and re-using existing code or remixing other programs.

To do this, students will need to possess the following skills and competencies:

	Algorithms	Data Structures	Modularity	Modelling & Abstraction	Debugging
Skill Unlocked	**Create an algorithm using a text-based language that includes functions, objects, conditionals, and arrays.**	**Build a data structure that can be used in an algorithm.**	**Create a module using a text-based language that can be applied in multiple programming contexts.**	**Evaluate existing models by identifying possible biases and using new data.**	**Explain how to test software for bugs and find solutions to problems they can anticipate.**
Almost There	Use loops, condition statements, and functions to create and reconstruct existing algorithms and improve their efficiency.	Explain how complex data structures (i.e., lists and maps) work and are used in algorithms. Describe their strengths and weaknesses.	Create a simple program using modules that already exist.	Create a model by observing, organizing, and analyzing patterns.	Analyze their own code and provide solutions to errors they've identified.
Keep Going	Create a simple algorithm to produce a certain action and describe other algorithms that can achieve the same result.	Explain how data structures (i.e., numbers and strings) work and how they are used to create algorithms. Describe their strengths and weaknesses.	Create modules by organizing a list of instructions into clusters using a visual programming language.	Identify relevant and irrelevant characteristics of objects and categorize them by their properties.	Solve errors in teacher-given text-based code.
Start Here	Provide the definition of an algorithm and explain its different components: loops, objects, condition statements, and functions.	Provide the definition of a data structure and its different functions: organizing, processing, retrieving, and storing data.	Recognize ways in which a large task can be broken down into a series of sub-tasks (i.e., modules).	Recognize patterns in the world and define common attributes of similar objects.	Debug programs built with a visual programming language.

For Bonus Points:

Document and justify computational processes when creating a program.

Describe the software development lifecycle.

Create more complex computer programs using more sophisticated algorithms and data structures.

Employ principles of writing maintainable code.

COMPUTING AND NETWORKS

By Grade 12 students should be able to identify an everyday task that can be automated or simplified using physical computing and design a solution.

To do this, students will need to possess the following skills and competencies:

	Hardware & Software	Connected Devices	Troubleshooting	Digital Connectivity	Cybersecurity
Skill Unlocked	**Apply a hardware and software solution to improve user experience.**	**Identify how devices obtain data from the environment and how data is shared between different devices.**	**Explain a technical problem and its solution in a way that allows others to implement the solution.**	**Describe how hierarchy and redundancy enhance the scalability and reliability of the Internet.**	**Define major cybersecurity risks and the types of solutions used to prevent cyberattacks.**
Almost There	Compare and contrast the differences between the various devices and software applications.	Describe how computing devices can be connected to other devices through physical or wireless connections to create a system.	Implement research skills to find solutions to problems with computers, external devices, or networks.	Explain the role of protocols as well as their importance in connecting billions of devices and analyze how to send and receive data efficiently.	Identify malicious content (i.e. spam, spyware, viruses, phishing, etc.) and implement practices that prevent or minimize their impact.
Keep Going	Describe how hardware and software work together (i.e. sending, receiving, processing, storing information as bits).	Compare and contrast the physical and non-physical aspects of computers and other types of devices.	Use troubleshooting strategies to solve common problems with computers, external devices, and networks.	Describe how information is sent and received through wireless and physical paths. Explain the role of routers and switches.	Explain the purpose of security software and how it protects a computer.
Start Here	Explain the composition of a computer system including physical and non-physical components.	Identify computing devices in our everyday lives and the ways in which people commonly use them.	Identify and describe common problems associated with computers, external devices, and networks.	Use the internet to conduct research and communicate with others.	Define cybersecurity and create safe passwords using effective criteria.

For Bonus Points:

Build a basic computer.

Build a robot from scratch.

DATA

By Grade 12 students should be able to apply storytelling techniques and/or visual graphics to explain data to reach conclusions, make decisions, or make predictions.

To do this, students will need to possess the following skills and competencies:

	Data and its Uses	Organizing Data	Assessing Information	Ownership & Governance	AI & Machine Learning
Skill Unlocked	**Identify possible risks of using data to make predictions about the world (i.e., bias, over-fitting, under-fitting).**	**Write formulas to create and organize new data.**	**Analyze and evaluate the purpose, motive, perspective, and bias of data and information.**	**Assess provincial and national, and data governance laws and regulations as well as Indigenous data governance agreements.**	**Assess how human biases are embedded within technical systems and artificial intelligence.**
Almost There	Formulate ways to use data to solve a problem or make a prediction. Describe the possible limitations of the solution or prediction.	Prepare an existing set of data so that it is easier to organize and process.	Interpret the purpose, motive, perspective, and bias of existing data and information.	Explain your data rights and create a plan for how to advocate for them.	Explain how machines learn. Discuss the specific ethical challenges with machine learning and AI.
Keep Going	Describe how data is used to make predictions about the world.	Organize simple data in a database or spreadsheet.	Analyze the accuracy of data and information found online by comparing it with other sources.	Discover who owns the data you produce online and identify whether that data is open or anonymized to remove your personal information.	Summarize the basic mechanics of AI systems and how data and machine learning interact.
Start Here	Identify methods of producing digital data.	Enter data into a database or spreadsheet.	Recall ways in which data and information can be produced by many different actors with different interests.	Identify ways to control and/or limit the types of data you share online.	List ways in which artificial intelligence and machine learning are being used in digital tools.

For Bonus Points:

Understand types of data structures and their advantages and disadvantages.

Make predictions using insight and knowledge gained from digitally processing and/or visualizing.

Assess the ethical implications of existing and potential technologies.

TECHNOLOGY AND SOCIETY

By Grade 12 students should be able to create a digital or physical artifact that examines specific positive and negative effects of technology on minorities, Indigenous peoples, and vulnerable groups as well as on the environment.

To do this, students will need to possess the following skills and competencies:

	Safe Communication	Ethics, Safety, & the Law	Tech & the Environment	History of Technology	Tech & Wellbeing
Skill Unlocked	**Evaluate how others use digital technologies to foster social inclusion and diverse needs.**	**Evaluate the effects of computer crime, hacking, virus distribution, and other illegal or unethical activities on society.**	**Assess how programming and computing can promote environmental sustainability.**	**Predict how the field of Computer Science might evolve and assess how work and future jobs might be affected.**	**Assess how specific technologies might be used to improve/impair our health and influence our behaviour.**
Almost There	Use digital technologies to interact with others. Identify methods of dealing with cyberbullying and hateful speech.	Compare and contrast software licensing models.	Identify methods to dispose of computing devices in a secure and environmentally safe way.	Analyze the evolution of computers and digital technologies.	Analyze potential health and mental health benefits and drawbacks of using technology.
Keep Going	Find examples of positive and inclusive online messaging, and how it impacts society. Identify cyberbullying and hateful speech, then explain how impacts society.	Apply basic copyright principles.	Explain the environmental impacts of producing computers and other digital technologies.	Describe examples of the earliest computers.	Recognize and reproduce aspects of an ergonomic workstation.
Start Here	Identify how computers have positively and negatively impacted how people communicate with each other.	Identify strategies to protect one's data and identity online.	Recognize environmental impacts of using digital technologies.	Identify key figures involved in shaping the history of Computer Science, focusing on the role of Canadians, women, and groups underrepresented in the tech sector.	Apply the school's technology policies.

For Bonus Points:

Describe how technology has influenced and shaped democracy.

Understand how policies governing technology and innovation are made and how they can shape the field of computer science, both positively and negatively.

DESIGN

By Grade 12 students should be able to create a multimedia artifact that combines three or more different content forms (i.e. text, audio, video, animations).

To do this, students will need to possess the following skills and competencies:

	User Design	Visual Design	Universal Design
Skill Unlocked	**Apply the principles of UX design to create a digital artifact and improve it through user testing.**	**Apply the principles of UI design to create a digital artifact that balances aesthetic design with practical application.**	**Design a user-friendly artifact that meets provincial and/or other known accessibility standards and accounts for a wide range of human diversity.**
Almost There	Explain how UX design creates an inclusive experience.	Identify the principles of effective UI Design.	Evaluate a digital artifact to assess whether it meets provincial and other known accessibility standards.
Keep Going	Prototype a digital artifact reflecting different user perspectives and needs.	Use the basic concepts of visual design to create content in any form (i.e.: text, audio, video, animation).	Describe a variety of adaptive technologies that help to improve accessibility. Identify strategies to design more inclusively.
Start Here	Identify the basic principles of design thinking.	Identify the basic principles of visual design.	Identify and explain the different elements of accessibility.

For Bonus Points:

Understand 3D design principles.

Create a 3D model.

Create code based art.

Explore the benefits and drawbacks of virtual reality.

HOW YOUR INPUT WILL HELP US

By completing our online survey or attending one of our online workshops, you can help us answer the following questions:

- Have we identified the right components of Computer Science education? What is missing?
- Have we identified the most appropriate competencies and skills for each of our five core components? What is missing?
- Are our proposed learning trajectories reasonable?
- Is there anything else we should consider including in our Framework?
- What tools and supports are needed to help educators implement these skills and competencies in their classroom?

Feedback submitted on this version of the document will be used to help refine our competence guide. This will ultimately help us to develop **the Pan-Canadian K-12 Computer Science Framework, which we'll launch in 2020.**

Questions? Send an email to: csframework@canadalearningcode.ca

THANK YOU FOR YOUR INTEREST IN STRENGTHENING COMPUTER SCIENCE EDUCATION IN CANADA