BIG IDEAS

The **design cycle** is an ongoing reflective process.	Personal design choices require self-exploration, collaboration, and evaluation and refinement of skills.	Tools and technologies can be adapted for specific purposes.

Learning Standards

Curricular Competencies	Content
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*** • Establish a point of view for a chosen design opportunity • Identify potential users, intended impact, and possible unintended negative consequences • Make inferences about premises and **constraints** that define the design space ***Ideating*** • Identify gaps to explore a design space • Generate ideas and add to others' ideas to create possibilities, and prioritize them for prototyping • Critically analyze how competing social, ethical, and sustainability considerations impact designed solutions to meet global needs for preferred futures • Work with users throughout the design process	*Students are expected to know the following:* • design opportunities • design cycle • **problem decomposition** • **structures** within existing code • ways to **modify** existing code to meet a particular purpose • **strategies** to predict effects of code modification • **pair programming** • programming language constructs to support input/output, logic, decision structure, and loops • **requirements** of a problem statement • **ways** to transform requirements into algorithms • translation of **design specifications** into source code • **tools** to aid in the development process • **pre-built libraries** and their **documentation** • inline commenting to document source code • **use of test cases** to detect logical or semantic errors • **computational thinking** processes • appropriate use of technology, including digital citizenship, etiquette, and literacy

Learning Standards (continued)

Curricular Competencies	Content
Prototyping • Identify and apply **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 the life cycle and evaluate its **impacts** • Construct prototypes, making changes to tools, materials, and procedures as needed • Record **iterations** of prototyping ***Testing*** • Identify feedback most needed and possible **sources of feedback** • Develop an **appropriate test** of the prototype • Collect feedback to critically evaluate design and make changes to product design or processes • Iterate the prototype or abandon the design idea ***Making*** • Identify appropriate tools, technologies, materials, processes, and time needed for production • Use **project management processes** when working individually or collaboratively to coordinate production ***Sharing*** • **Share** progress while creating to increase opportunities for feedback • Decide on how and with whom to share or promote their product, creativity, and, if applicable, **intellectual property** • Consider how others might build upon the design concept • Critically reflect on their design thinking and processes, and identify new design goals • Assess ability to work effectively both as individuals and collaboratively while implementing project management processes	

Learning Standards (continued)

Curricular Competencies	Content
Applied Skills • Apply safety procedures for themselves, co-workers, and users in both physical and digital environments • Identify and assess skills needed for design interests, and develop specific plans to learn or refine them over time **Applied Technologies** • Explore existing, new, and emerging tools, **technologies**, and systems to evaluate their suitability for their design interests • Evaluate impacts, including unintended negative consequences, of choices made about technology use • Analyze the role technologies play in societal change • Examine how cultural beliefs, values, and ethical positions affect the development and use of technologies	

Big Ideas – Elaborations	APPLIED DESIGN, SKILLS, AND TECHNOLOGIES – Computer Programming Grade 11

- **design cycle:** includes updating content, tools, and delivery. The design process can be non-linear.

Curricular Competencies – Elaborations	APPLIED DESIGN, SKILLS, AND TECHNOLOGIES – Computer Programming Grade 11

- **user-centred research:** research done directly with potential users to understand needs and requirements
- **constraints:** limiting factors, such as available technology, expense, environmental impact, copyright
- **sources of inspiration:** may include experiences, users, experts, and thought leaders
- **information:** may include professionals as experts, secondary sources, collective pools of knowledge in communities and collaborative atmospheres both online and offline
- **impacts:** including the social and environmental impacts of extraction and transportation of raw materials, manufacturing, packaging, transportation to markets, servicing or providing replacement parts, expected usable lifetime, and reuse or recycling of component materials
- **iterations:** repetitions of a process with the aim of approaching a desired result
- **sources of feedback:** may include peers; users; First Nations, Métis, or Inuit community experts; other experts and professionals both online and offline
- **appropriate test:** includes evaluating the degree of authenticity required for the setting of the test, deciding on an appropriate type and number of trials, and collecting and compiling data
- **project management processes:** setting goals, planning, organizing, constructing, monitoring, and leading during execution
- **Share:** may include showing to others, use by others, giving away, or marketing and selling
- **intellectual property:** creations of the intellect such as works of art, invention, discoveries, design ideas to which one has the legal rights of ownership
- **technologies:** things that extend human capabilities

- **problem decomposition:** subdivide a problem into manageable, self-contained tasks
- **structures:** for example, key elements such as variables, functions, use of Whitespace
- **modify:** for example, altering values of variables, parameters of a function or loop
- **strategies:** hand tracing code, guess and test (experimentation)
- **pair programming:** two programmers work together at one workstation. One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The two programmers switch roles frequently.
- **requirements:** a complete set of requirements that will support the rest of the software development cycle without the need to revisit the problem statement in the future
- **ways:** for example, pseudocode, iterative refinement, flowcharts, UML, other design entities
- **design specifications:** for example, pseudocode, algorithms, flow charts, unified modeling language (UML)
- **tools:** for example, integrated development environment (IDE), computer language appropriate for problem/project
- **pre-built libraries:** for example, external libraries for graphical user interfaces or gaming, sensor libraries for hardware such as coding devices
- **documentation:** interpretation of library documentation/application programming interface (API)
- **use of test cases:** for example, running test cases to compare expected versus actual output and printing the value of variables to aid in the debugging process
- **computational thinking:** formulating problems and their solutions so they are represented in a form that can be solved through an algorithmic process. Key components are decomposition, patterns and generalizations, abstraction, and algorithmic thinking.