

## BIG IDEAS

Decomposition and **abstraction** help us to solve difficult problems by managing complexity.

**Algorithms** are essential in solving problems computationally.

Programming is a tool that allows us to implement **computational thinking**.

**Solving problems** is a creative process.

**Data representation** allows us to understand and solve problems efficiently.

## Learning Standards

Curricular Competencies	Content
<p><i>Students are expected to do the following:</i></p> <p><b>Reasoning and modelling</b></p> <ul style="list-style-type: none"> <li>• Develop <b>fluent, flexible, and strategic thinking</b> to analyze and create algorithms</li> <li>• Explore, <b>analyze</b>, and apply mathematical ideas and computer science concepts using <b>reason, technology, and other tools</b></li> <li>• <b>Model</b> with mathematics in <b>situational contexts</b></li> <li>• <b>Think creatively</b> and with <b>curiosity and wonder</b> when exploring problems</li> </ul> <p><b>Understanding and solving</b></p> <ul style="list-style-type: none"> <li>• Develop, demonstrate, and apply conceptual understanding through experimentation, <b>inquiry</b>, and problem solving</li> <li>• <b>Visualize</b> to explore and illustrate computer science concepts and relationships</li> <li>• Apply <b>flexible and strategic approaches</b> to <b>solve problems</b></li> <li>• Solve problems with <b>persistence and a positive disposition</b></li> <li>• Engage in problem-solving experiences <b>connected</b> with place, story, cultural practices, and perspectives relevant to local First Peoples communities, the local community, and other cultures</li> </ul>	<p><i>Students are expected to know the following:</i></p> <ul style="list-style-type: none"> <li>• <b>access variables</b> in memory</li> <li>• ways in which <b>data structures</b> are organized in memory</li> <li>• <b>uses</b> of multidimensional arrays</li> <li>• classical algorithms, including <b>sorting and searching</b></li> <li>• use of Big-O notation to help predict run-time <b>performance</b></li> <li>• <b>recursive problem solving</b></li> <li>• <b>persistent memory</b></li> <li>• <b>encapsulation</b> of data</li> <li>• ways to <b>model mathematical problems</b></li> </ul>

Learning Standards (continued)

Curricular Competencies	Content
<p><b>Communicating and representing</b></p> <ul style="list-style-type: none"> <li>• <b>Explain and justify</b> computer science ideas and <b>decisions</b> in <b>many ways</b></li> <li>• <b>Represent</b> computer science ideas in concrete, pictorial, and symbolic forms</li> <li>• Use computer science and mathematical vocabulary and language to contribute to <b>discussions</b> in the classroom</li> <li>• Take risks when offering ideas in classroom <b>discourse</b></li> </ul> <p><b>Connecting and reflecting</b></p> <ul style="list-style-type: none"> <li>• <b>Reflect</b> on mathematical and computational thinking</li> <li>• <b>Connect mathematical and computer science concepts</b> with each other, other areas, and personal interests</li> <li>• Use <b>mistakes</b> as <b>opportunities to advance learning</b></li> <li>• <b>Incorporate</b> First Peoples worldviews, perspectives, <b>knowledge</b>, and <b>practices</b> to make connections with computer science concepts</li> </ul>	