**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**

Students are expected to do the following:

**Reasoning and modelling**
- Develop fluent, flexible, and strategic thinking to analyze and create algorithms
- Explore, analyze, and apply mathematical ideas and computer science concepts using reason, technology, and other tools
- Model with mathematics in situational contexts
- Think creatively and with curiosity and wonder when exploring problems

**Understanding and solving**
- Develop, demonstrate, and apply conceptual understanding through experimentation, inquiry, and problem solving
- Visualize to explore and illustrate computer science concepts and relationships
- Apply flexible and strategic approaches to solve problems
- Solve problems with persistence and a positive disposition
- Engage in problem-solving experiences connected with place, story, cultural practices, and perspectives relevant to local First Peoples communities, the local community, and other cultures

**Content**

Students are expected to know the following:

- access variables in memory
- ways in which data structures are organized in memory
- uses of multidimensional arrays
- classical algorithms, including sorting and searching
- use of Big-O notation to help predict run-time performance
- recursive problem solving
- persistent memory
- encapsulation of data
- ways to model mathematical problems
### Learning Standards (continued)

<table>
<thead>
<tr>
<th>Curricular Competencies</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Communicating and representing</strong></td>
<td></td>
</tr>
<tr>
<td>• Explain and justify computer science ideas and decisions in many ways</td>
<td></td>
</tr>
<tr>
<td>• Represent computer science ideas in concrete, pictorial, and symbolic forms</td>
<td></td>
</tr>
<tr>
<td>• Use computer science and mathematical vocabulary and language to contribute to discussions in the classroom</td>
<td></td>
</tr>
<tr>
<td>• Take risks when offering ideas in classroom discourse</td>
<td></td>
</tr>
<tr>
<td><strong>Connecting and reflecting</strong></td>
<td></td>
</tr>
<tr>
<td>• Reflect on mathematical and computational thinking</td>
<td></td>
</tr>
<tr>
<td>• Connect mathematical and computer science concepts with each other, other areas, and personal interests</td>
<td></td>
</tr>
<tr>
<td>• Use mistakes as opportunities to advance learning</td>
<td></td>
</tr>
<tr>
<td>• Incorporate First Peoples worldviews, perspectives, knowledge, and practices to make connections with computer science concepts</td>
<td></td>
</tr>
</tbody>
</table>