Coding in MTH1W - Read, Alter and Write

CEMC: Bringing Teachers Together Virtually
August 2022

Marcel te Bokkel
marcel.tebokkel@yrdsb.ca
@te_bokkel

Today’s Deck:
Coding in MTH1W

» Or really in any course
» Goals
  » survey Ontario coding curriculum
  » brief look pseudocode
  » use Scratch to read, write and alter code
  » view collection of resources
### Progression of Coding Expectations from Grade 1-8

<table>
<thead>
<tr>
<th>Grade 1</th>
<th>Grade 2</th>
<th>Grade 3</th>
<th>Grade 4</th>
<th>Grade 5</th>
<th>Grade 6</th>
<th>Grade 7</th>
<th>Grade 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential events</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential and concurrent events</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, and repeating events</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, repeating, and nested events</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves conditional statements and other control structures</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves conditional statements and other control structures</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves conditional statements and other control structures</td>
<td>C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves conditional statements and other control structures</td>
</tr>
<tr>
<td>C3.2 read and alter existing code, including code that involves sequential events, and describe how changes to the code affect the outcomes</td>
<td>C3.2 read and alter existing code, including code that involves sequential and concurrent events, and describe how changes to the code affect the outcomes</td>
<td>C3.2 read and alter existing code, including code that involves sequential, concurrent, and repeating events, and describe how changes to the code affect the outcomes</td>
<td>C3.2 read and alter existing code, including code that involves sequential, concurrent, repeating, and nested events, and describe how changes to the code affect the outcomes</td>
<td>C3.2 read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the code</td>
<td>C3.2 read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the code</td>
<td>C3.2 read and alter existing code, involving the analysis of data in order to inform and communicate decisions, and describe how changes to the code affect the outcomes and the efficiency of the code</td>
<td>C3.2 read and alter existing code, involving the analysis of data in order to inform and communicate decisions, and describe how changes to the code affect the outcomes and the efficiency of the code</td>
</tr>
<tr>
<td>What is it?</td>
<td>Sequential events</td>
<td>Add: concurrent events</td>
<td>Add: repeating events</td>
<td>Add: nested events</td>
<td>Add: conditional statements</td>
<td>Efficiency</td>
<td>Counting and subprograms</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------</td>
<td>------------------------</td>
<td>-----------------------</td>
<td>-------------------</td>
<td>-----------------------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>What does that mean?</td>
<td>One instruction after another</td>
<td>Instructions happening at the same time</td>
<td>Repeating instructions</td>
<td>Nested means an instruction within an instruction</td>
<td>Conditional means making decisions based on something (if… then…)</td>
<td>Efficiency</td>
<td>Counting and subprograms</td>
</tr>
</tbody>
</table>
#### Grade 9 Expectations

**Overall Expectation**

C2 apply coding skills to represent mathematical concepts and relationships dynamically, and to solve problems, in algebra and across the other strands

<table>
<thead>
<tr>
<th>C2.1</th>
<th>C2.2</th>
<th>C2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>- use coding to demonstrate an understanding of algebraic concepts including variables, parameters, equations, and inequalities</td>
<td>- create code by decomposing situations into computational steps in order to represent mathematical concepts and relationships, and to solve problems</td>
<td>- read code to predict its outcome, and alter code to adjust constraints, parameters and outcomes to represent a similar or new mathematical situation</td>
</tr>
</tbody>
</table>
Task Bank Slide Layout

» Pseudocode
» Read Code
» Alter Code
» Write Code
Pseudocode

Specific Expectation
C2.2 create code by decomposing situations into computational steps in order to represent mathematical concepts and relationships, and to solve problems

Computational steps, could also be Pseudocode, consists of short English phrases used to explain specific tasks/steps within a program.

Ideally, pseudocode does not include keywords exclusive any specific computer language.

Pseudocode should be written as a list of consecutive phrases; we may also draw arrows to show direction or clarify repeating processes.
Making A Peanut Butter Sandwich
Decomposing situations into computational steps
Pseudocode

Think about making a detailed plan as to how to sequence the steps in order to solve the problem.

In what order would we put the blocks to determine the average of 5 numbers that have been entered?
Start the code

Start a Loop of 5 times

Input a number

Combine the input with the total sum

Display the average

Assign total sum divided by 5 to variable called average

End the loop

End the code
In what order would we put the blocks to determine the average of 5 numbers that have been entered?

Did your pseudocode look different?
Read Code

Which one doesn't belong?
Justify your thinking.
Read the code shown and describe its function.
DeBug it! Is a number positive or negative?

**Problem**: Find the error in the “Determining whether a number is negative or positive” problem

Read the code shown and describe its function.

**REFLECTION PROMPTS**
- What was the problem
- How did you identify the problem?
- How did you fix the problem?
- Did others have alternatives approaches to fixing the problem
Getting Started: What can you make with these 10 blocks?

Create a project using only these 10 blocks.

Use them once, twice, or multiple times, but use each block at least once.

START HERE

- Test ideas by experimenting with each block
- Mix and match blocks in various ways
- Repeat!
Scratch Resources

» MIT Scratch website
  » Create an account
  » Registering for a Teacher’s Account (takes a few days)
    » Teacher Account Guide

» CS First

» Split screen
Write Code

For the equation of the line $y = 2x + 5$, determine whether or not an entered coordinate is located on the line.

C1.2 create algebraic expressions to generalize relationships expressed in words, numbers, and visual representations, in various contexts.
Write Code

For the equation of the line \( y = 2x + 5 \), determine whether or not an entered coordinate is located on the line.

C2.2 create code by **decomposing situations** into computational steps in order to represent mathematical concepts and **relationships**, and to **solve problems**
Is the point on the line?

Think about how to sequence the steps in order to solve the problem.

1. Input an x-value
2. Start the code
3. If results are equal display “on”
4. Compare 2x+5 with y
5. Input a y-value
6. If results are not equal display “not on”
7. End of Code
8. Input an x-value
For the equation of the line $y = 2x+5$, determine whether or not an entered coordinate is located on the line.

C2.1 use coding to demonstrate an understanding of algebraic concepts including variables, parameters, equations, and inequalities
**Is the point on the line?**

**Problem:** For any equation of a line in the form \( y = ax + b \), determine whether or not an entered coordinate is on that line.

Use **pseudocode** to detail the plan that could be used to code this problem.

**Read** the code shown and describe its function.

**Modify** the code shown below to determine if the coordinate is on, above or below the line.

**Write** the complete code for the above problem.
MTH1W Coding Resources

» MTH1W task bank linked to expectations and individually organized by -
  » Pseudocode
  » Read Code
  » Alter Code
  » Write Code

» Other Resources
  » Intro to Scratch (grades 1 to 8)
  » Sample Programs
  » Coding and Spreadsheet Ideas

Questions?