SCIENCE NORTH SCIENCE NORD
SUDBURY, ONTARIO, CANADA

Be efficient	Grade 5 and 6

Lesson Plan

Coding Tool	Scratch, MakeCode
Time Required	3 periods

Math Curriculum Connections

Algebra: Coding

C3. solve problems and create computational representations of mathematical situations using coding concepts and skill

Specific Expectations

C3.1 solve problems and create computational representations of mathematical situations by writing and executing code
C3.2 read and alter code and describe how changes to the code affect the outcomes

Financial Literacy; Money and Finance

C3. demonstrate the knowledge and skills needed to make informed financial decisions

Specific Expectations

F1.5 calculate unit rates for various goods and services, and identify which rates offer the best value

Science Curriculum Connections

Conservation of Energy and Resources

- Conservation is one way of reducing the impacts of using energy and resources

Overall Expectations

- Analyse the immediate and long-term effects of energy and resource use on society and the environment, and evaluate options for conserving energy and resources

Specific Expectations

1.2 evaluate the effects of various technologies on energy consumption
3.1 identify a variety of forms of energy and give examples from everyday life of how that energy is used

Description

Students will explore the concept of efficiency and how it applies to both energy and coding. With this parallel in mind, students will have the chance to alter given code to make it more efficient. They will then use MakeCode, a free, open-source coding platform, to program sensors in a Micro:bit to create efficient technology that could be used in a home.

Success Criteria

- The 5th and 6th grade students will be able to define efficiency in terms of fuel efficiency, home efficiency and coding efficiency
- The 5th and 6th grade students will be able to debug code and find more efficient alternatives by finding patterns and using loops
- The 5th and 6th grade students will be able to create efficient code using Microsoft Makecode and the Microbit

Materials and Media

- Pencils
- Calculators
- Devices with web access
- Coding Efficiency Handout & Answers
- Energy Efficiency Handout & Answers
- Fuel Consumption Handout & Answers
- Debugging Handout & Answers
- Coding Handout

Computational Thinking Skills

This lesson introduces the concept of an efficiency as it applies to computer programming. In programming, efficient code uses the fewest lines of code to accomplish a function. This is beneficial as it makes debugging and review much simpler. Identifying sequences and patterns is very useful in this type of efficiency.

As efficient code makes it easier to read and debug code, this lesson will also look at debugging some existing code, specifically to help make it more efficient. Programmers rarely work along, so having easy to follow code makes it easier for other programmers to find and fix potential issues. Students will have the chance to fix existing code that doesn't work or alter code to improve its efficiency.

The debugging process is a skill set that can be used in all coding situations. Another essential skill for programmers is being able to write pseudocode. Pseudocode is the process of writing out code offline, to help plan for what will be input with the computer. Using pseudocode helps make code more efficient as it eliminates potential errors before investing too much time into the programming language. Students will have the opportunity to write code in this lesson to see how good planning can lead to more efficient and better written code.

The final coding element in this lesson is use the program Microsoft MakeCode. Students will use conditional statements to compare different values and use it to create smart technology for a home.

Introduction

The term efficiency or being efficient can be applied in different contexts. When something is said to be efficient, it successful in producing a desired result. More specific to what this lesson focuses on, true efficiency is producing the desired result with little to no waste. The overarching theme of this lesson is efficiency. Efficiency is an essential skillset when coding, so this lesson will look at what efficiency means in different contexts to help demonstrate how important it is to not be wasteful. By looking at efficiency in terms of fuel consumption and energy efficient homes, students will identify that we try to reduce waste in most of what we do, and that coding is not different.

Action

Unplugged Activities

This lesson has three different unplugged activities, each of which look at efficiency in different ways. By completing these three activities, students will have a better understanding of how efficiency in coding can be valuable, just as it is in the two real world examples. Each activity has a handout and an answer key.

1) **Fuel consumption**: This first exercise has students embarking on a fictitious road trip where they will need to cost out the price of fuel before going. They will compare the cost of fuel for different vehicles based on their posted fuel consumption ratings from Natural Resources Canada. In addition to efficiency concepts, this activity gives

practice manipulating numbers with operators in an everyday life scenario. As such this activity can fit as part of the financial literacy curriculum.

2) **Energy efficient homes and appliances:** The second exercise looks at energy efficiencies within a fictitious home. Here, we will contrast energy conservation with energy efficiency. Students will perform calculations on energy savings for different everyday scenarios.

3) **Coding efficiencies**: This exercise has students identifying patterns and creating loops with pseudocode. Their task will be to write an efficient algorithm for taking care of a fictitious pet. After creating a comprehensive list of the steps needed to take care of a pet over the course of year, students will fit the items into nested loops so that the tasks are performed at the appropriate frequency. For example, a feeding task might occur daily for some pets. For others, it could occur bi-weekly. After the tasks are assigned a time interval, the loops will have to be nested into each other so that the code runs properly. The most efficient code can have tasks repeating without having to write them more than once.

Debugging Activity

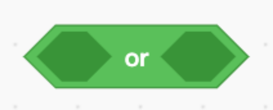
The next portion of the lesson has students debug inefficient code, which can be found with the ***Debugging Handout***. The students will receive code that is written in an inefficient manner. It has long repeating sequences. Students will be tasked with identifying and re-grouping the sequences to be made more efficient. They can use the space next to the code to write out their ideas for the code. The three tasks that they will have to debug are:

- Debug 1 – On and Off Again: https://scratch.mit.edu/projects/474054351/
- Debug 2 – High Score: https://scratch.mit.edu/projects/474041754/
- Debug 3 – Making Shapes: https://scratch.mit.edu/projects/474063834/

After they have identified their edits and debugs, they can access the inefficient code online in Scratch. There they will have the opportunity to rewrite the code efficiently and test it out. The links for the three activities are posted above. Adding the online portion allows them to test their code. As students work through their tasks, have them test their code often to make sure their new efficient code follows exactly what the long code did. If you haven't remixed another project before, be sure to check out this remix guide: https://en.scratch-wiki.info/wiki/Guide_to_Remixing

Solving the first activity, the blinking light, only requires students to identify the pattern and put the repeating elements in the loop.

The second activity, changing score, is trickier because the pattern isn't as obvious. What will help students is knowing that there is an “or” block.



Writing the simplified pseudocode, this gives:

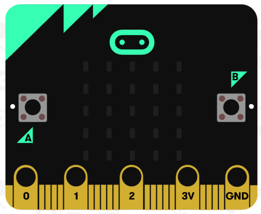
- if the sprite is touching the apple or the banana or the orange, change score by +1.
- If the sprite is touching the cake or the donut or the muffin, change score by –1.

The third activity, making an octagon, is the trickiest piece to debug so we’ve broken it down for you here. When looking at the handout, you’ll see a ‘move 50 steps’ block repeated as well as a ‘point in direction X’ where X changes on an interval of 45 each time. The trick with making this program more efficient is removing the ‘point in direction blocks’ and repeating them with a ‘turn 45 degrees’ block. With this in place, it’s much easier to see the pattern and students can make a more efficient loop that repeats the move and turn blocks.

More efficient code for each of the three activities is included with the ***Debugging Activity Answers***.

Coding Math Activity

This activity will use Microsoft MakeCode to program a BBC Micro:bit. MakeCode is a free, open-source coding platform that can be used to create and run programs on actual hardware or in the absence of the hardware, a simulated target. In this activity, the Micro:bit will be the simulated target. A Micro:bit is a popular microcontroller with several built-in inputs and outputs that can be programmed to execute different functions. If you have access to a class set of Micro:bits, you can incorporate them into the lesson. If not, you can use the simulator in the coding platform.



Since this lesson discusses energy, it’s worth discussing the different forms of energy that can be found with a Micro:bit. Inputs or sensors measure energy and, depending on what we have coded, use energy from the battery to create an output:

Power Source

- Batteries: Store chemical energy and transform it into electrical energy

Sensors

- Accelerometer: detects movement based on changes in gravitational potential energy
- Temperature sensor: detects heat energy
- Light sensor: detects light energy

- Digital Compass: detects magnetic fields
- Buttons: detects electrical energy (require some muscular energy to activate)
- Touch sensor(logo): detects changes in capacitance (changes when you touch it)

Outputs

- 25 LEDs: transform electrical energy into light
- Speaker: transforms electrical energy into sound

Given all this information, students will be able to complete tasks that take advantage of the different forms of energy that can be coded for with the Micro:bit. One place we see the smart use of technology in terms of energy is in a home. We call this home automation or also smart homes. We can choose different limits on how the things that use power will behave to add elements of efficiency. The two tasks using the Micro:bit simulate elements that might be found in an energy efficient smart home:

Task 1: Use a sensor to program a more efficient light - Students will take code from a light that is always on and edit it so that the light only turns on when it is dark. If time allows, students can try editing their code again to add different conditions that might add more efficiency.

Task 2: Use a sensor to turn on the heat or air in a home. Since the micro:bit doesn't include a heater or air conditioner, students will choose a different output as stand-ins for these. The Coding Handout uses lights as substitutes. They will first program a simple thermostat that controls only a heater. Students will have to consider their temperature threshold. The next edit of their program will consider both a heater and an air conditioner. What range of temperatures will be comfortable? Would they have different code for different times of day or different times of year? How do we keep these devices from doing more work than they have to? Have them think about conditions that would increase efficiency.

The full code for both these programs can be found in the ***Coding Handout***.

Closure and Assessment

Upon completion of the lesson, there are multiple opportunities for assessment, primarily among the three unplugged activities and the debugging activity. The handouts for each can be collected and used for assessment of learning, with the data collected going towards the math strand of a student's achievement grade. Alternatively, these handouts as well as the work done with the Micro:bit can be used for assessment as learning to gain feedback and check for next steps in regard to the '*conservation of energy*' section of the curriculum.

Adaptations	Extensions
• Microsoft MakeCode has options to use the physical Micro:bit or its online emulator • Students can modify pre-existing code instead of starting from scratch to facilitate their coding • Allow students to use a calculator in the financial literacy part of the lesson	• Both tasks in the coding portion have options to expand on the code and make it more technical with the Micro:bit • Challenge students to code a unique smart-home gadget using some of the different sensors or outputs on the Micro:bit • Encourage efficiency when creating the code; challenge students to use the fewest lines of code as possible for all activities