MATH NOW

Print Version Sampler

Online Version available

Ontario

Trillium List Approved

MATH NOW Grades 1– 8

Grade 1	Grade 2	Grade 3	Grade 4
Chapter 1 ***(Sample included)*** What is Quantity?	**Chapter 1** Count Me In	**Chapter 1** Numbers, Numbers Everywhere!	**Chapter 1** Number Sense
Chapter 2 Information Station	**Chapter 2** From Sorts to Graphs	**Chapter 2** Data in My Life	**Chapter 2** Data Literacy
Chapter 3 Patterns All Around	**Chapter 3** Patterns Please	**Chapter 3** A World of Patterns!	**Chapter 3** Patterns and Relationships
Chapter 4 Quantities to 50	**Chapter 4** We Value Numbers	**Chapter 4** By the Hundreds!	**Chapter 4** Addition and Subtraction
Chapter 5 Moving Through Space and Time	**Chapter 5** On the Move	**Chapter 5** On the Move!	**Chapter 5** Location and Movement
Chapter 6 Sharing Quantities Fairly	**Chapter 6** Together or Apart?	**Chapter 6** Fair Play	**Chapter 6** Multiplication and Division
Chapter 7 Balancing Act	**Chapter 7** Balancing Quantities	**Chapter 7** Equivalent Relationships	**Chapter 7** ***(Sample included)*** Variables, Equations and Inequalities
Chapter 8 All Shapes and Sizes	**Chapter 8** Shapes All Around	**Chapter 8** I Spy Shapes	**Chapter 8** Geometric Reasoning
Chapter 9 Take a Chance!	**Chapter 9** By Any Chance	**Chapter 9** In All Likelihood	**Chapter 9** Probability
Chapter 10 Understanding Sequence in Coding	**Chapter 10** Writing Instructions in Coding	**Chapter 10** ***(Sample included)*** Making Loops in Coding	**Chapter 10** Using Variables in Coding

Grade 5	Grade 6	Grade 7	Grade 8
Chapter 1 Number Sense	**Chapter 1** Number Sense	**Chapter 1** ***(Sample included)*** Number Sense	**Chapter 1** Number Sense
Chapter 2 Data Literacy	**Chapter 2** Data Literacy	**Chapter 2** Data Literacy	**Chapter 2** Data Literacy
Chapter 3 Patterns and Relationships	**Chapter 3** Patterns and Relationships	**Chapter 3** Patterns and Relationships	**Chapter 3** Patterns and Relationships
Chapter 4 Addition and Subtraction	**Chapter 4** Addition and Subtraction	**Chapter 4** Part-whole Relationships	**Chapter 4** Working with Rational Numbers
Chapter 5 Location and Movement	**Chapter 5** Location, Transformations, Measurement	**Chapter 5** Two-Dimensional Shapes and Measurement	**Chapter 5** Geometric Relationships and Movements
Chapter 6 Multiplication and Division	**Chapter 6** Multiplication and Division	**Chapter 6** Working with Rational Numbers	**Chapter 6** Multiplying and Dividing Rational Numbers
Chapter 7 Variables, Equations and Inequalities	**Chapter 7** Variables, Equations and Inequalities	**Chapter 7** Variables, Equations and Inequalities	**Chapter 7** Variables, Equations and Inequalities
Chapter 8 Geometric Reasoning	**Chapter 8** Geometric Reasoning	**Chapter 8** Three-Dimensional Shapes and Measurements	**Chapter 8** Geometric Reasoning and Measurement
Chapter 9 Probability	**Chapter 9** Probability	**Chapter 9** Probability	**Chapter 9** Probability
Chapter 10 Introducing Conditional Statements in Coding	**Chapter 10** ***(Sample included)*** Enhancing Your Skills in Coding	**Chapter 10** Writing Subprograms in Coding	**Chapter 10** Making Decisions in Coding

LESSON 3 Describing Five

Learning Goal

We can recognize and count quantities of 5.

Introduction

Numbers have quantity. We read and write numbers using symbols and words.

Using symbols	0	1	2	3	4	5
Using words	zero	one	two	three	four	five

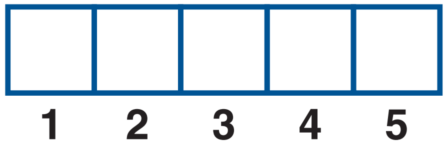
Zero or 0 represents nothing or no quantity.

Tell something about yourself using each number.

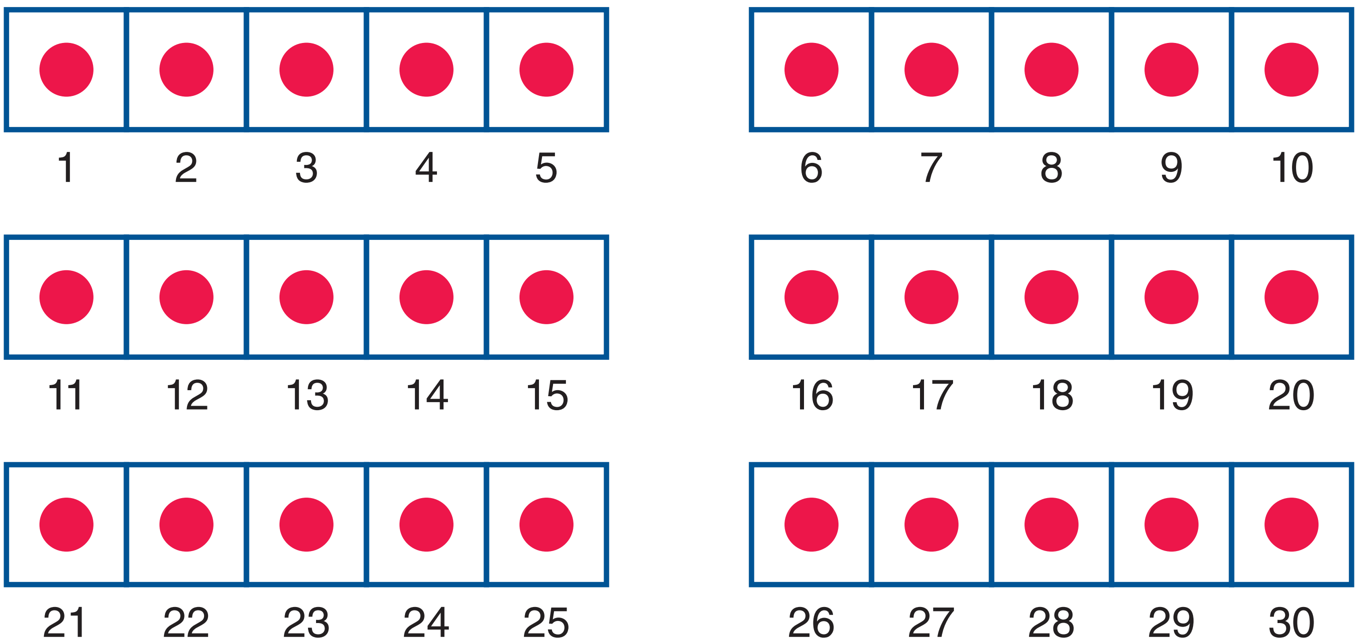
Use the Count and Write to 5 worksheet to practise counting and writing numbers.

Exploring **Fun with 5**

How much is **5**?

This is called a **five-frame**  because it has 5 squares.

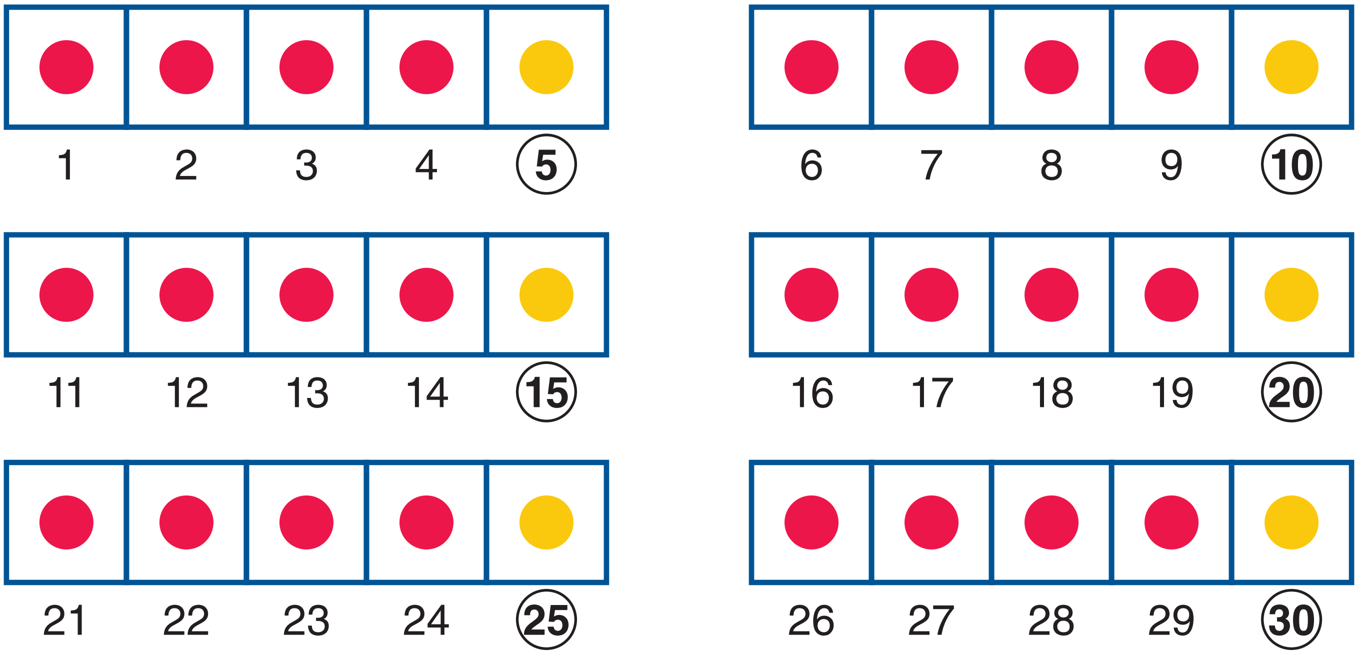
Counting by 1s



- Point to the number as you count each counter.
- Start at 30. Count back to 1.
- Talk about any number patterns you see.

Counting by 5s

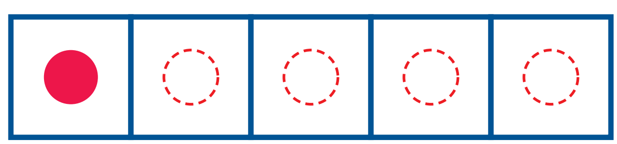
Counting by 5s means we only say every fifth number.



- Start at 5 and count **forward** by 5s.
- Start at 30 and count **back** by 5s.
- Talk about the **number patterns** you see.

Math Clinic — Fun with Five-Frames

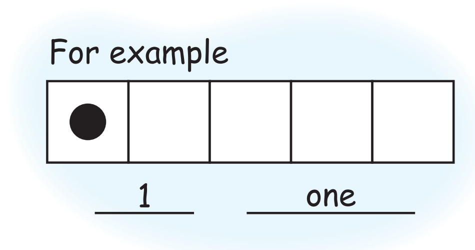
Here is 1 counter on a five-frame.



How many more counters are needed to make 5?

Use the Fun with Five-Frames worksheet to record how many counters are on each five-frame.

Think about how many counters you need on the empty squares to make 5 each time.

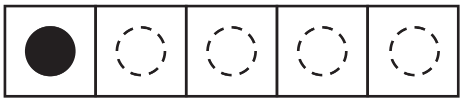


Practising Fundamentals

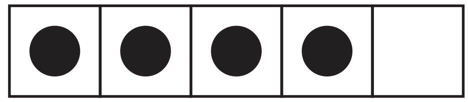
Making 5

Use 5-frames and counters to help you.

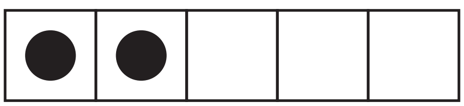
How many more counters are needed to make 5?

1. 

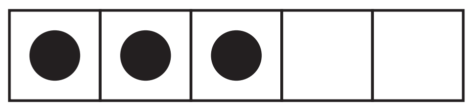
5 is 1 and ___4___ more.

2. 

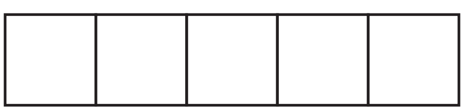
5 is 4 and ______ more.

3. 

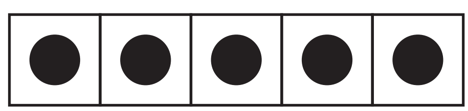
5 is 2 and ______

4. 

5 is 3 and ______

5. 

5 is ______ and 0

6. 

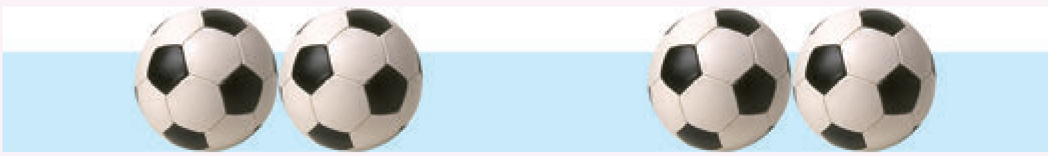
5 is ______ and 5.

Problem Solving

The Same or Different Quantities?

1. a) How many soccer balls?

   b) How many here?



   c) How many here?

   d) How many here?

2. Use linking cubes to **make 5** in as many **different ways** as possible.

   Record your findings using pictures, words and numbers.

3. From memory, tell a partner how many ways you can make 5.

Reflecting

What I Know About 5

- Count 30 small objects **by 5s** onto a table. Spread the objects apart. How many objects are there now? Move them to other locations on the table. **Without counting**, do you know how many objects there are each time? Explain.
- What strategy helped you to count by 5s? Explain why. Describe another strategy that you might try next time.

LESSON 3 Inequalities

Learning Goal

We can solve inequalities using addition and subtraction, and verify and graph the solutions.

We understand the relationship between spending and saving to meet a financial goal.

Introduction

Andre De Grasse is a world-class Canadian sprinter. He is the first Canadian sprinter to win three medals in one Olympics, which he did in 2016. In 2020, he won Canada's first Olympic gold medal in sprinting since 1996.

Sprinters run a short distance, like 100 m or 200 m, at the fastest speed they can run. Sprinting is a track-and-field sport. As of 2022, the world record in men's 100 m sprinting is 9.58 seconds, and in women's 100 m sprinting, it is 10.49 seconds.

How fast can you sprint?

Qualifying Time

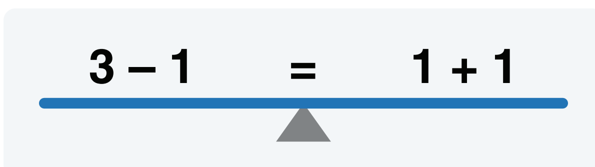
Suppose your school is selecting sprinters to participate in a track and field event. The qualifying time for 100 metres is 16 seconds.

Look at the following number line and discuss how it can help you answer the question, “How do you know if a student has qualified?”

You will learn more about scenarios like this in this lesson.

Exploring Understanding Inequalities

In Lesson 2, you learned how to write and solve algebraic equations using various models including the **balance model** to represent the mathematical concept of equivalence.



When the values of the two sides of the equation are equal, the equation is balanced. We can isolate and determine the unknown value in the equation by adding or subtracting equal values on both sides of the equation.

Can the balance model do more?

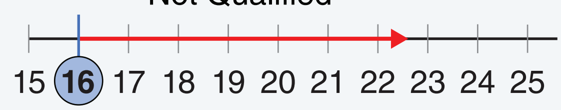
Part 1 – Inequality Signs

Review the symbols from **BLM-B53: Inequality Symbols**. Remember that the greater value will be lower on the balance scale. Inequalities are math statements used to show the relationship between two expressions or values that are not equal. An inequality can include an equal component such as **less than or equal to (≤)** and **greater than or equal to (≥)**.

Write mathematical statements for the selection criteria in the Qualifying Time scenario in the Introduction.

Part 2 – Inequalities and Number Lines

Use the number line presented in the Introduction to illustrate each of the mathematical statements you write and give an example, such as, you are not qualified if your time is greater than 16 seconds ($t > 16$), i.e., 17, 18 and so on, including numbers between the whole seconds.



On a number line, show how the symbols "≥" and ">" are represented using a filled-in dot and a circle.

Math Clinic **Solving Inequalities**

BLM-B54: Graphing Inequalities on Number Lines shows the four inequality signs and how each sign is represented on a number line.

Let's solve the following problem using number lines to show the solutions.

In a fundraising event, you and your friend set a goal of raising \$20. Your friend raised \$7.

How much money will you have to raise, a) to exceed the goal?

b) to meet or exceed the goal?

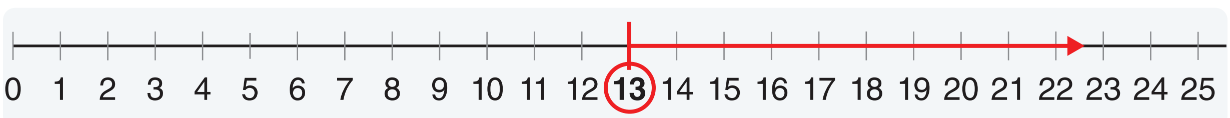
a) The inequality that represents 'exceeding the goal' is a + \$7 > \$20, where a is the amount of money that you must raise.

You are comparing the values of the expression on the left side of the inequality with the value \$20 on the right side.

Isolate a using the balance model as you did when solving algebraic equations in Lesson 2.

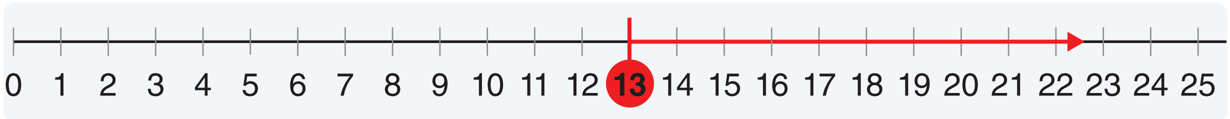
- Subtract \$7 from both sides (Your friend raised \$7, and you need to raise more than \$20 – \$7 to exceed the target.)
- a + \$7 **– \$7** > \$20 **– \$7**
- a > \$13

You must raise more than \$13 to exceed the goal of \$20. This can be represented on a number line as



b) To meet or exceed the fundraising goal, the inequality is $a + \$7 \geq \$20$.

You must raise \$13 or more to exceed your fundraising goal. This can be represented on a number line as



Solve the inequality, $y + 9 > 16$.

Show your strategy, verify your solution and graph it on a number line.

Practising Fundamentals — Working with Inequalities

1. Circle the incorrect solution for z in $z \leq 7$.

   A 0 B 3 C 7 D 8

2. Write an inequality for each statement.

   a) p is greater than 4

   b) z is greater than or equal to 10

   c) 11 is less than b

   d) 15 is less than or equal to x

3. Is the inequality True or False for the value given?

   a) The value for y is 10. The inequality is $y - 3 > 6$.

   b) The value for p is 18. The inequality is $13 > p - 5$.

   c) The value for h is 5. The inequality is $16 < h + 9$.

4. Write the inequality for situation.

   a) Naomi has at least 10 guppies in her aquarium.

   b) Aidan has more guppies than Naomi.

   c) Ali lost count of how many fish he has in his aquarium. He thinks he has either 15 or fewer than 15.

5. Draw the number line that graphs the inequality $c > 4$.

6. Draw the number line that graphs the inequality $7 \leq t$.

Problem Solving

Creating a Financial Plan

There are many examples of mathematical inequalities in our lives. One of these occurs when we are making a financial plan to meet a goal. What do you think is the purpose of a financial plan?

In this task, you will work in small groups to create a plan to meet a financial goal.

There many decisions to make, and situations sometimes happen that can affect the outcome.

In your group, decide on a situation and financial goal that you want to meet, such as buying an item, saving to make a donation, etc.

- Describe a financial goal ≤ 20 (set as a variable).
- Decide how long you have to meet the goal (set as a variable).
- Identify situations where money can be earned.
- Predict how much money is saved from money earned.
- Identify situations that might require you to spend money that you saved.
- Identify assumptions that you may have.

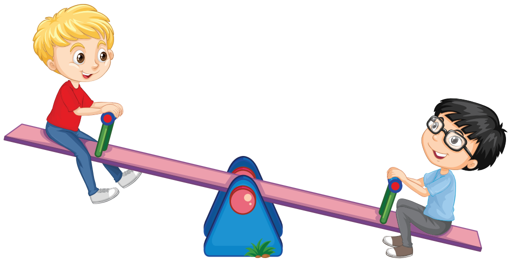
In your financial plan, apply your understanding of inequalities. Here are some things to think about as you discuss and work on this task.

- Include a goal, as a variable, to show how changes of circumstances may affect the outcome.
- Use numbers ≤ 20 in your plan.
- Create expressions to represent different parts of the financial plan such as the earnings, expenses, etc.
- Use inequality signs to relate the different parts of your plan.
- When solving inequalities, show your work, verify your solution, and graph it using a number line.

Be prepared to explain your plan to others.

Reflecting What I Know About Inequalities

- Use your own words to finish this sentence "An inequality is like a seesaw because …."



- What are examples of inequalities that can occur in real-world financial contexts?

LESSON 3 Factors

Learning Goal

We can determine the greatest common factor for a variety of whole numbers up to 144 and the lowest common multiple for sets of two or three whole numbers.

Introduction

A building can be composed of storeys or floors. Floors are composed of apartments. Apartments are composed of rooms.

You have learned that numbers can be composed of smaller numbers.

If you relate the way a building is composed and the factoring of a number, what would prime factors be? Why?

Prime numbers have **exactly** two factors, 1 and the number itself; for example, 7 has two factors, 1 and 7, (1 × 7 = 7).

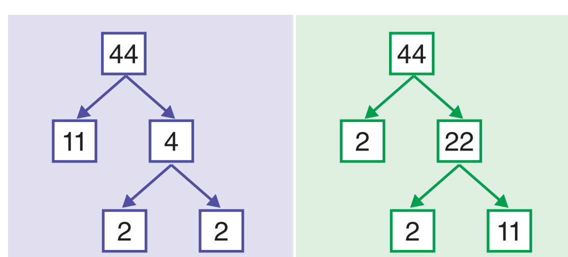
Composite numbers are composed of more than two factors; for example, 9 has 3 factors, 1, 3, and 9 (1 × 9 = 9 and 3 × 3 = 9).

Factoring is part of the reverse process of multiplication, and you use it to identify the factors of a number.

Ella and Dean want to determine the factors of the number 44.

They each use a factor tree to find the factors. Their results are shown in the diagrams. Examine Ella and Dean's factor trees.

- Is one or both of their factor trees correct?



Ella's Factor Tree

Dean's Factor Tree

Be prepared to explain your reasoning to the class.

Ella and Dean express the prime factorization of the number 44 as the following multiplication sentence: $2 \times 2 \times 11 = 44$. Explain why.

Exploring Greatest Common Factor and Lowest Common Multiple

The life cycles of a species of cicada insects called periodical cicadas occur at intervals of 13 years and 17 years. What is special about these numbers?

One early theory is that two life-cycle time periods means periodical cicadas would not all emerge at the same time to compete for the same food. They would only emerge together every 221 years!

A current theory is that their life cycles have evolved to avoid their predators and parasites. This enables enough cicadas to live, lay eggs and survive as a species.

- How do scientists know that the life cycles of periodical cicadas only emerge every 221 years?

First let's learn more about **factoring**.

With a partner, choose two different numbers from 100 to 144.

- One number should end with the digit 2, 4, 6 or 8.
- The other number should end with a digit of 5 or 0.

Each of you choose a strategy to factor your two numbers.

When you have both finished, compare your factors for the first number.

- Are they identical or did you have some differences?
- Did you identify the same prime factors? Explain why or why not.

Write the prime factors for one of your numbers as a multiplication sentence.

Compare your prime factors.

- Does the order of the prime numbers matter in your multiplication sentence?

Repeat the process for your other number.

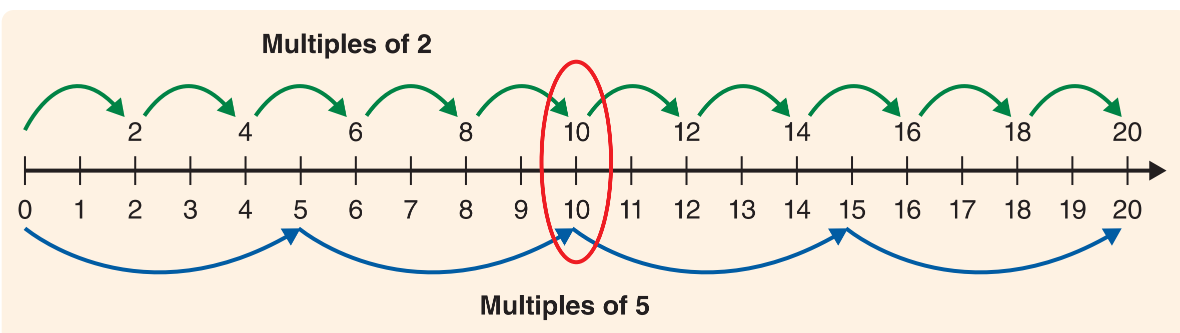
Let's compare your set of two numbers, their factors and multiplication sentences.

- Are there any factors that are common for both of the two numbers?
- Which of those common factors is the biggest?

This number is called the **greatest common factor** or **GCF** of your two numbers. The greatest common factor is the largest whole number that divides evenly into all numbers in a given set. As a self-check, you can divide both original numbers by this GCF. The GCF should divide evenly with no remainder.

Let's identify the multiples of a set of numbers and determine the **lowest common multiple** or **LCM** for the numbers. The lowest common multiple is the smallest whole number that the numbers in a set can divide into evenly.

Examine this number line and compare the multiples of 2 and 5.



- What are the common multiples of 2 and 5?
- Which of these common multiples is the smallest?

We see that 10 and 20 are multiples of both 2 and 5. The LCM is 10 because it is the smallest whole number that divides evenly by both 2 and 5 ($10 \div 2 = 5$ and $10 \div 5 = 2$).

Recall that the periodical cicadas have life cycles of 13 years and 17 years.

- How do you think scientists determined these cicadas only emerge together every 221 years? Show your reasoning.

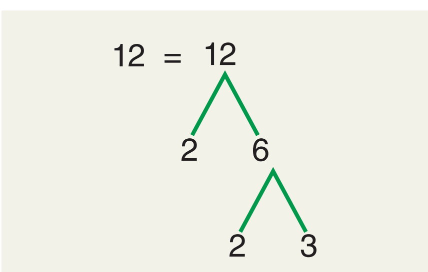
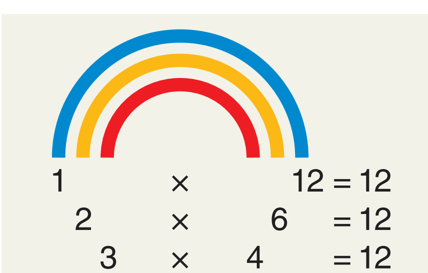
Math Clinic Reviewing Factors and Multiples

Prime Factorization

- Prime factorization identifies the prime numbers that multiply together to make the original number. Different strategies can be used to determine the prime numbers.
- The results of a prime factorization can be expressed as a multiplication sentence. For example, the prime factorization of 12 can be expressed as
$2 \times 2 \times 3 = 12$ or it can be written using exponents as
$2^2 \times 3 = 12$.

Two Strategies to Determine the Factors of Numbers

The factors of 12 can be represented by:

<table>
<tr>
<td>Factor Tree

</td>
<td>Start with the number 12.
• Pick a factor, for example, 2, and form one branch from 12.
• Divide 12 by 2 to give 6, forming the other branch.
• Continue until all the prime factors are found.
• $12 = 2 \times 2 \times 3$
• This is prime factorization.</td>
</tr>
<tr>
<td>Factor Rainbow

</td>
<td>• Start with the smallest number and find the pair of numbers that multiply together to give 12.
• There is no number between 3 and 4, so we have completed the process.
• 1, 2, 3, 4, 6, 12 are the factors of 12.</td>
</tr>
</table>

Greatest Common Factor (GCF) and Least Common Multiple (LCM)

- The GCF for a set of numbers is the largest number that divides evenly into all numbers in that set.
- The LCM for a set of numbers is the smallest number that can be evenly divided by each of the numbers in that set.

Multiplication Facts and Factors

Knowing your multiplication facts makes factoring come naturally.

×	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144

Examine this 12 × 12 multiplication chart. What do you notice?

- Pick any even number from 2–144. In which columns and rows could it appear?
- Pick any number from 5–140 with the last digit a 5 or 0. In which columns or rows does it appear?
- How is knowing multiplication facts related to determining factors?

Practising Fundamentals

Determining GCFs and LCMs

1. List the multiples of 13. List the multiples of 7.

   a) What numbers do the lists have in common?

   b) Identify the lowest common multiple. Show how you know this is the smallest multiple of both 13 and 7.

2. Use different strategies to identify the factors of 54 and 72. Show your work.

   a) What is the greatest common factor of 54 and 72? How do you know?

   b) Determine the least common multiple of 54 and 72. Justify your answer.

3. A number has exactly eight factors, including itself and 1. Two of its factors are 22 and 55. What is the number? Show your work.

4. When listening to a beat in music class, Jona claps every fourth beat, and Kyle claps every sixth beat.

   a) When would Jona and Kyle clap at the same time? Show your work.

   b) A third student, Marci, joins them and claps every eight beats. How often would all three students clap at the same time? Show your work.

   c) What is the greatest common factor of 4, 6, and 8?

5. James goes to the gym every 4 days. Simon goes to the same gym every 3 days. They just met. When will they see each other again?

6. Is this statement true or false? $2 \times 7 \times 10 = 140$ is a prime factorization. Justify your answer.

Problem Solving

How many in the bucket?

Annie's restaurant specializes in chicken wings. Customers request them by the bucket. The most popular size is Biggest Bucket O'Wings.

Use the information below to determine how many wings are in the Biggest Bucket O'Wings.
Show your work.

- If the wings are removed two at a time, one wing will be left.
- If they are removed three at a time, two wings will remain.
- If the wings are removed four, five or six at a time, then three, four and five wings, respectively, will remain.
- If they are taken out seven at a time, no wings will be left.

What is the smallest possible number of wings that could be in the bucket?

Explain how you know.

Be prepared to present to the class.

Reflecting

What I Know about Greatest Common Factor and Lowest Common Multiple

- Work with an elbow partner and create the steps to determine the GCF and LCM of 30, 36, and 60.
- Now that you have learned more about factoring, where do you think it is used in your life? Explain, using an example.

MATH NOW CODING

MATH NOW CODING ALIGNMENT OF SCRATCH BLOCKS								
Grade	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Curriculum Expectations: **C3.1 write and execute code involving** **C3.2 read and alter code involving**	**Sequential Events**	**Concurrent Events**	**Repeating Events**	**Nested Events**	**Control Structures**	**Efficient Code**	**Subprogram**	**Decision-Making**
Introduction								
Events								
Code								
Sequence								
Sprite								
Scratch Coding Panes								
Coordinate Plane (Scratch Stage)								
Animation								
Costumes								
Backdrop								
Scratch Blocks								
Motion								
move ( ) steps								
turn clockwise ( ) degrees								
turn counter-clockwise ( ) degrees								
go to (random position)								
go to x: ( ) y: ( )								
glide ( ) secs to ( )								
glide ( ) secs to x: ( ) y: ( )								
point in direction ( )								
change x by ( )								
set x to ( )								
change y by ( )								
set y to ( )								
if on edge, bounce								
set rotation style ( )								
Looks								
say ( ) for ( ) seconds								
say ( )								
think ( ) for ( ) seconds								
think ( )								
switch costume to ( )								
next costume								
switch backdrop to ( )								
next backdrop								
change size by ( )								
set size to ( ) %								
change (colour) effect by ( )								
set (colour) effect to ( )								
show								
hide								
Sound								
play sound ( ) until done								
start sound ( )								
stop all sounds								
change ( ) effect by ( )								
set ( ) effect to ( )								
clear sound effects								
change volume by ( )								
set volume to ( ) %								
Events								
when green flag clicked								
when ( ) key pressed								
when this sprite clicked								
when backdrop switches to ( )								
when I receive (message)								
broadcast ( )								
broadcast ( ) and wait								

MATH NOW CODING ALIGNMENT OF SCRATCH BLOCKS								
Grade	1	2	3	4	5	6	7	8
Curriculum Expectations: **C3.1 write and execute code involving** **C3.2 read and alter code involving**	Sequential Events	Concurrent Events	Repeating Events	Nested Events	Control Structures	Efficient Code	Subprogram	Decision-Making
Scratch Blocks								
Control								
wait ( ) seconds								
repeat ( )								
forever								
if < > then								
if < > then else								
wait until < >								
repeat until < >								
stop ( )								
Sensing								
touching ( )								
touching colour ( )								
ask ( ) and wait								
key ( ) pressed								
Operators								
( ) + ( )								
( ) - ( )								
( ) * ( )								
( ) / ( )								
pick random ( ) to ( )								
( ) > ( )								
( ) < ( )								
( ) = ( )								
< > and < >								
< > or < >								
not < >								
join ( ) ( )								
letter ( ) of ( )								
length of ( )								
( ) contains ( )								
( ) mod ( )								
round ( )								
(10^) of ( )								
Variables								
Make a Variable								
set ( ) to ( )								
change ( ) by ( )								
show variable ( )								
hide variable ( )								
Make a List								
My Blocks								
Make a Block								
Pen								
erase all								
stamp								
pen down								
pen up								
set pen colour to ( )								
change pen (colour) by ( )								
set pen size to ( )								

LESSON 1 Algorithms

Learning Goal

We can use algorithms to program computers to perform tasks and solve problems.

Introduction

Let's find out how we can get a computer to work for us! Coding is easy to learn and is fun!

In earlier grades, you learned how to write steps in a sequence for someone to perform a task or solve a problem. Now, you will write the steps to get a computer to perform a task or solve a problem.

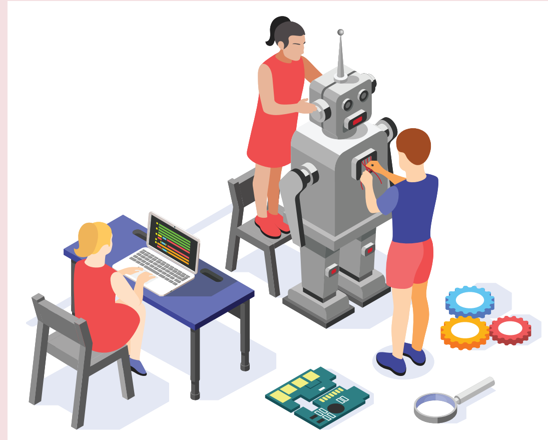
Understanding the following terms is key to learning how to code.

A **command** is an instruction that tells a computer what to do.

Coding is how you write the commands. **Code** refers to a set of commands that computers can follow. Sometimes we use the term **script** to mean a set of commands.

A computer **application** is a set of codes written to instruct the computer to perform a specific task. In this lesson, **program** is also used to mean application.

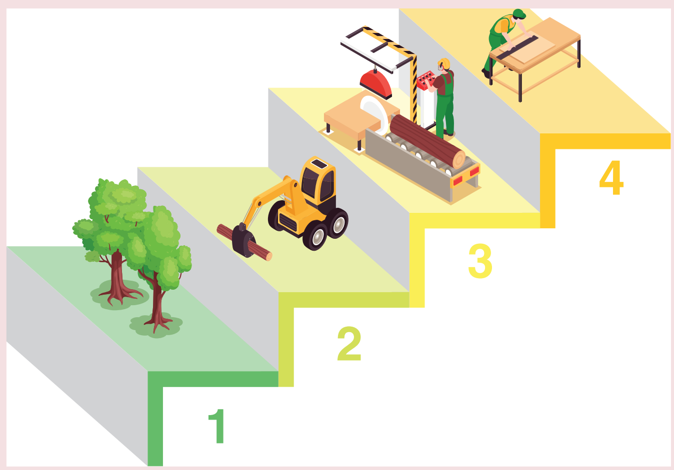
You **execute** the applications when you start, run or process an application in a computer.



Exploring All About Algorithms!

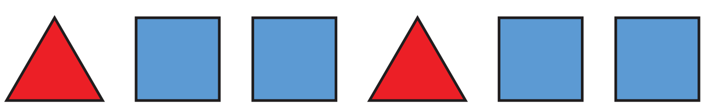
An **algorithm** is a set of sequential steps used to perform a task or to solve a problem.

If everyone uses the same algorithm to do a task, they will all get the same result.



You have been using algorithms to solve number problems, organize and display data, and create patterns.

We can follow the algorithm on the next page to make this pattern.



The arrowheads (↓) indicate the order of the events or steps.

Set of Commands

1. Draw one red triangle.

↓

2. Draw two blue squares.

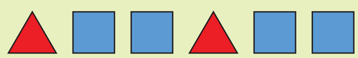
↓

3. Repeat Step 1 and Step 2.

↓

Output

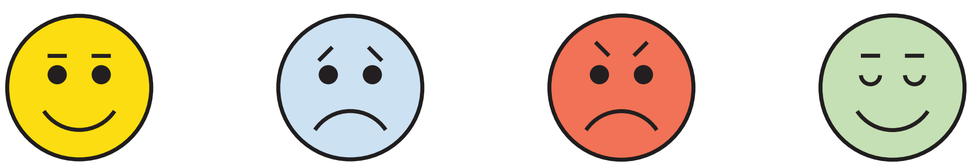
The result is a pattern of 1 red triangle, 2 blue squares.



Drawing Emojis

Work with a partner.

- Choose one of the emojis shown.



Each of you will create a set of instructions to draw that emoji.

- Write your instructions vertically in proper order to show the process of drawing the emoji.
- Your instructions are a set of rules. You have created an algorithm to draw an emoji!
- Trade algorithms with your partner. Follow your partner's instructions to draw the emoji.

Do you get the same output as your partner's chosen emoji?

If not, talk about what changes may be needed.

Computer Language

We communicate with computers using languages they understand.

Computers do not understand the instructions you wrote to draw an emoji.

Coding is done using programming language.

> **Programming language** is a set of commands that the computer understands and can use to perform a task or solve a problem.

We use programming language or computer language to translate our instructions or commands into codes that computers can understand.

Block-based coding is a programming language that we will be using.

Math Clinic Blocks of the Day

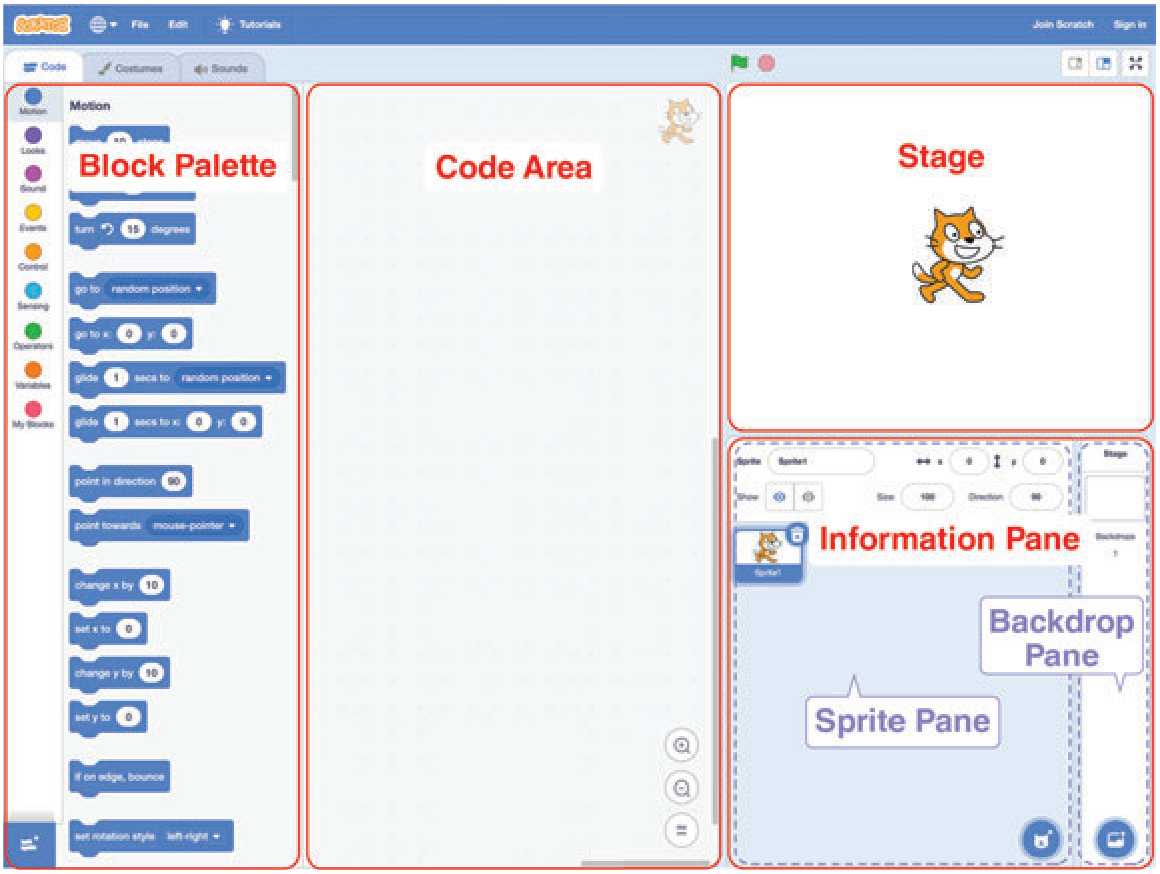
Massachusetts Institute of Technology (MIT) developed **Scratch**. It is a website that we can use to learn coding. Scratch allows you to drag and drop coding blocks into a coding area and piece them together to create your code.

A **sprite** is a character used in computer graphics. When the code is executed, the actions are done by the sprite. Every sprite in a project has its own code.

Let's review the **Scratch Coding Pane** as shown on the next page.

There are four divisions.

- The **Block Palette** has the coding blocks you can use. Clicking on a coloured circle such as Motion or Looks will show the coding blocks for that category. The coding blocks look like jigsaw puzzle pieces.
- The **Code Area** is where you build your code. You will drag and drop the coding blocks from the Block Palette into the Code Area. You can execute your code by clicking on the blocks or a stack of blocks in this area.

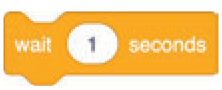


- The **Stage** shows the background of the project and the sprites. When a program is executed, the output is the action of the sprites, and it is shown in the Stage.
- The **Information Pane** has two sections—**Sprite Pane** and **Backdrop Pane**.

  The **Sprite Pane** gives information about the sprites used, such as the symbol or icon for each sprite. If your program uses more than one sprite, the Code Area is open for the sprite selected each time.

  The **Backdrop Pane** lets you choose a background image for your project from the Scratch Backdrops Library, or you may create an image to upload to your project.

Let’s review some coding blocks that you explored before.

Motion Blocks	
	The **move ( ) steps** block moves the sprite forward. It is called a stack block, and has a notch at the top and a bump below, like a jigsaw puzzle piece. If you find a block that fits these features, you can stack them.
	A quarter turn can be written as a 90 degree turn or 90° turn. The first **turn ( )** block turns the sprite clockwise The next block turns the sprite counterclockwise. If you set “90 degrees,” the sprite turns one quarter of a full turn. The **turn ( )** blocks are also stack blocks.
Control Blocks	
	The **wait ( ) seconds** block pauses the script for the specified time before the next command is executed. It is a stack block. You can stack it with other blocks that fit.
Events Blocks	
	The **green flag** block is used to start a script or set of events. It has a rounded top. So, you cannot place any blocks on top of it. You can stack a block with a notch at the top below the **green flag** block.
	The **when ( ) key pressed** block is also used to start a script. You can choose the key you want to start the events in the drop-down list.

Practising Fundamentals

Code Words

1. Describe what will happen if you click on each of these blocks.

   a) move 10 steps

   b) turn ↻ 90 degrees

   c) turn ↺ 90 degrees

Do parts a) and b) below for each of questions 2 to 6.

a) Draw the block code.

b) Create a Scratch Project to check how the sprite responds with your answer.
(You can use one project for all the questions.)

For example, to move the sprite 20 steps forward, turn 90° clockwise and move 50 steps forward, you should have the following.

a) drawing

move 20 steps
turn ↻ 90 degrees
move 50 steps

b) using this in your Scratch project

move 20 steps
turn ↻ 90 degrees
move 50 steps

2. How do you move the sprite forward 100 steps?

3. How do you tell the sprite to make a half turn clockwise?

4. How can you tell the sprite to make a quarter turn clockwise and move 30 steps?

5. How can you change the code in Question 4 by using wait ( ) seconds and move ( ) steps blocks to slow down the movement?

6. How can you use an Events block to execute the code in Question 5?

Problem Solving

Write Some Code

1. Create a new Scratch Project.

Let's practise using more blocks to see how they work.

a) Drag and drop the `move ( ) steps` block into the Code Area.

Enter different values for the number of steps. Click on the block several times for each value. What is the outcome for the sprite in the Stage area?

Use the mouse to drag and move the sprite around in the Stage.

b) Drag and drop both of the `turn ↻ ( ) degrees` and `turn ↺ ( ) degrees` blocks into the Code Area.

Keep them separate. (Do not stack them.)

Click on each of these blocks several times and discuss the outcomes for the sprite.

Change the value of both `turn ↻ ( ) degrees` and `turn ↺ ( ) degrees` blocks to "90." Click on each block several times and discuss the outcomes.

c) Stack the `wait ( ) seconds` block in between two `move ( ) steps` blocks.

Click on the stack and discuss the outcome for the sprite.

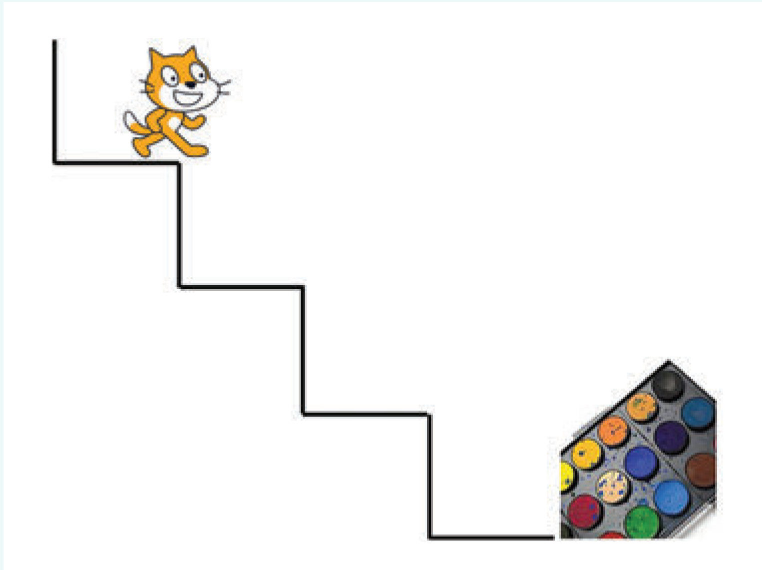
Change the value of the `wait ( ) seconds` block and click on the stack. What happens?

Remove this block and keep the two `move ( ) steps` blocks stacked.

Click on the stack and discuss the outcome for the sprite.

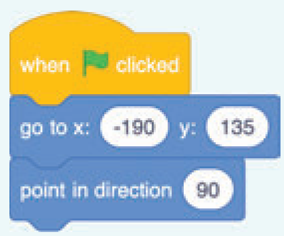
Are you ready to write some code? Let's get coding!

2. Work in small groups to write code that makes the sprite go down the stairs to get the paint set.



Your teacher will provide you with the link to access **Scratch Project Write Some Code**. You will remix the project and save it.

When you click on "See inside," you will see



This code is done for you so that every time you click on the green flag above the Stage, the sprite will reset to the top of the stairs.

The 

block is an Events block that starts the command blocks stacked below it.

Start a new stack of code separate from the "green flag" stack already there by doing the following.

- Drag and drop the Events block when [▾] key pressed into the Code Area.
- Click the ▽ for a drop-down list to select the key you want to execute your code.

You should be ready to code now! Here are some tips to help you.

- turn ↺ ( ) degrees and turn ↻ ( ) degrees blocks with a value of "90 degrees" is a quarter turn.
- You can use a value of "90" for your move ( ) steps blocks.
- When you want to test your program, remember to reset the sprite and then execute your code using your selected key.

Reflecting — What I Know About Coding

- Someone asks you, "What is coding?" What will be your explanation?
- Choose one of the following and answer in your math journal.
  - » Describe a time during this lesson when you were frustrated with how things were going. How did you work your way through that frustration?
  - » Describe a time during this lesson when you knew 'you got it!'
    How did you feel?
    How did that feeling help you during the rest of the lesson?
  - » Describe a time when you felt happy/unhappy with how things were going with the rest of your group. (no names, just how you felt and how you worked it out)

LESSON 2 Debugging

Learning Goal

We can write and execute code and debug the applications. We can read and alter existing code and describe how changes to the code affect outcomes.

Introduction

When computer programs or software do not work the way that they are supposed to, coding specialists must test the applications and find the errors. The errors must be fixed before the application goes live. Software and coding programs are human-made. The probability of having errors in the code is high, but like all mistakes we make, we can learn from our coding errors to improve the efficiency of the code.

We call errors that occur in a computer program 'bugs.' Debugging or removing errors is part of the coding process.

Determination, persistence and critical thinking skills are necessary to solve the types of problems that coders face when they are debugging programs.

Before attempting the debugging assignments in this lesson, think about a time when you encountered a problem.

- How did you solve the problem?
- How did you feel after you solved the problem?
- What steps did you take to ensure that the problem would not occur again?
- When you failed to solve the problem, what did you do?

Exploring Debugging

Your teacher will provide you with the link **Scratch Project Debugging**. Open and execute the code.

Something is wrong. The application should show the correct payment after you enter the tank capacity of your vehicle and the gas grade, regular or premium.

However, the calculation and the error message are not correct.

Work with your group and analyse the code. Record any questions that you may have, for example, note any coding blocks you have not used before or blocks that you do not understand and so on. Be ready to share your questions and thinking with the rest of the class.

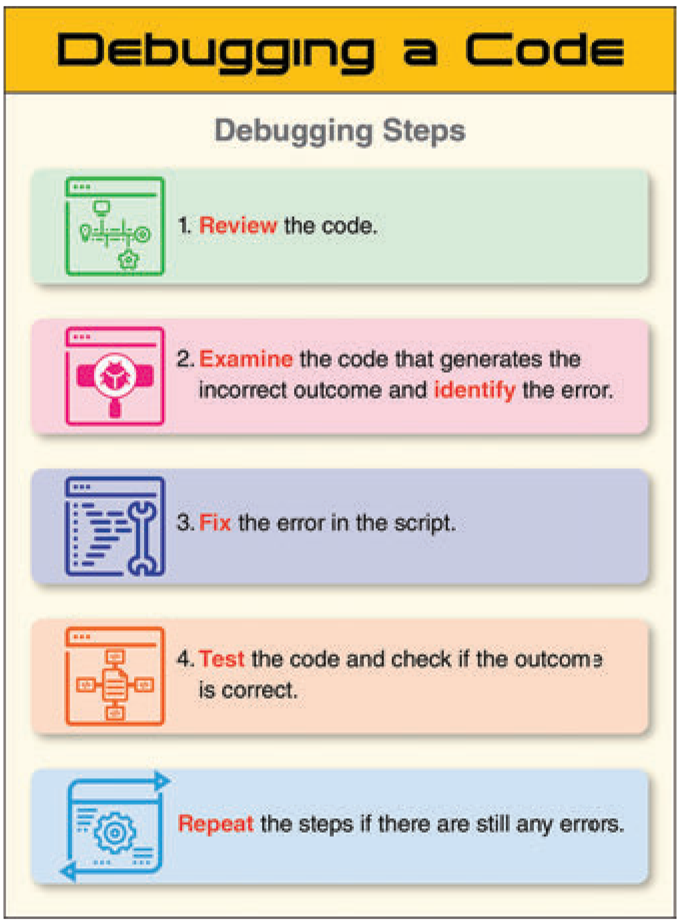
Return to your group and try debugging the code.

Debugging Method

Several incorrect coding blocks may be used in an application or they may be used more than once.

1. As you analyse the code, **fix one error at a time in the order of the script**.
2. Test the code after each correction so that you know if that error was fixed and to help you identify the next error (if there is one).
3. Do not attempt to fix all the errors at once.

Use the method on **CBLM 6.2: Debugging a Code** poster to plan how you work.



Your teacher will show you the suggested code after you finish debugging. You can compare your findings with the suggested code.

Math Clinic Blocks of the Day

You have used Boolean blocks in evaluating conditional statements. Here are two more Scratch blocks you can learn to use in your codes.

<table>
<tr><th colspan="2">Sensing</th></tr>
<tr><td>
</td><td>The touching ( ) block (and the block below) are Boolean blocks, which are hexagonal shapes and are used with the conditional blocks to evaluate a condition. The control statement will then return a value that is either “true” or “false.”

These Boolean blocks are used with conditional blocks, if < > then and if < > then else to perform different commands depending on the evaluation result.

The touching ( ) block has 3 options in the dropdown list:
mouse-pointer – controls the sprite using your mouse
edge (of the Stage) – controls the sprite going outside the Stage
another sprite – The names of the other sprites show in the dropdown list. You can write code to respond when the sprite is touching (or not touching) another sprite.</td></tr>
<tr><td>
</td><td>The touching colour ( ) block works like the block above.

When the sprite is touching the selected colour, the conditional block returns a value of “true,” otherwise it returns a value of “false.”

Click on the colour slider to choose the colour and set the saturation and brightness values to 100% for the time being.</td></tr>
</table>

Practising Fundamentals

Conditional Statements

1. Create Scratch code to have the sprite move left and right forever. Remember that you can use the 

block

2. Create code to use the mouse to catch a sprite moving randomly. Stop the code when you catch the sprite.

3. Create code to set two sprites moving at random and stop the code when they collide.

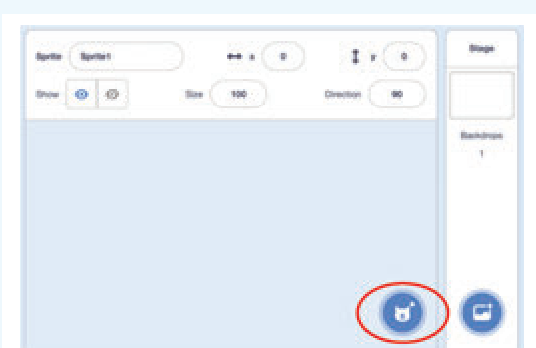
4. Create code to move the sprite randomly for 10 seconds. Catch the sprite and score one point. Output the score on the Stage.

Problem Solving

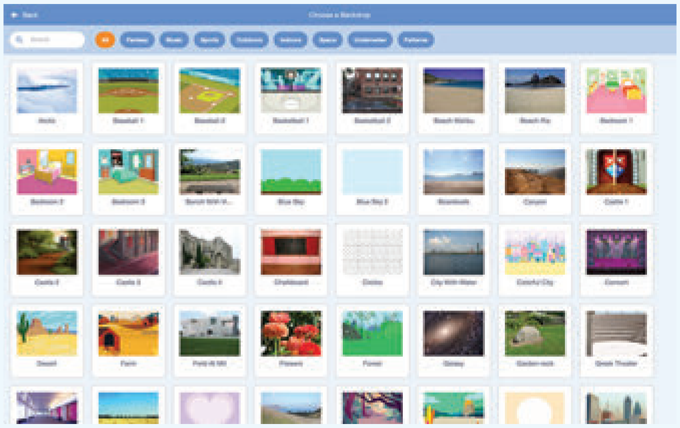
Park Your Car!

In Grade 4 Chapter 10 Lesson 1 Coordinate Plane, you learned how to find the coordinate grid backdrop in Scratch and upload it to your Scratch project. In Scratch, you can choose different backdrops to use on the Stage. The Scratch backdrop library offers a number of images that you can use. You can also create your own backdrops for your projects. The uploading process is similar to uploading a sprite image discussed in Lesson 1 Animation.

Click the Choose a Backdrop icon in the Information Pane.



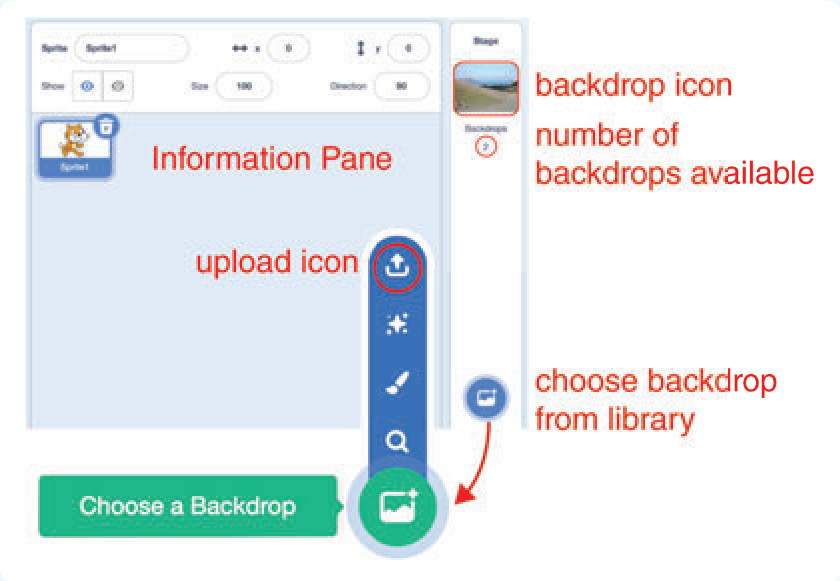
The screen will display all the images in the backdrop library. Select your backdrop, and it will show on the Stage.



You can click the Backdrops tab at the top to see all the backdrops you have selected down the left side. Choose the one you want to use.

Click your sprite and the Code tab to return to your coding screen.

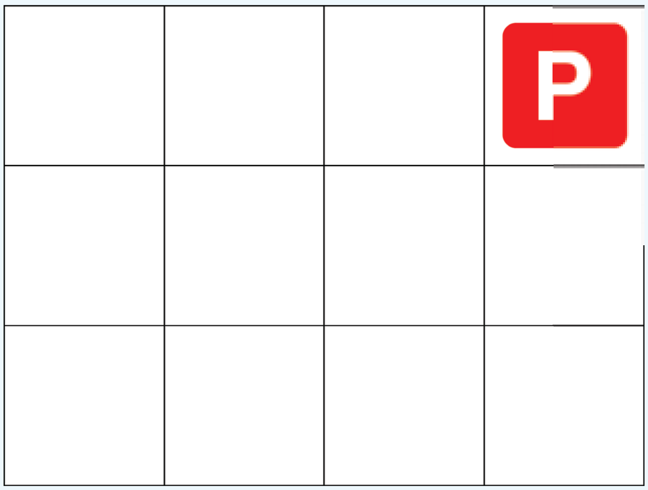
If you want to upload an image from your computer, click the upload icon and find the image on your computer.



Follow the procedure of uploading a sprite in the Animation lesson.

The dimension of the Scratch Stage is 4:3. You may want to use a resolution of 960 × 720 if you want to create a backdrop that occupies the entire screen.

Here is a parking lot grid that you can use as a backdrop. You can click and download the Parking Grid to your computer and then upload it to a Scratch Project to see how it appears in the background.



Your teacher will provide you with the link to access **Scratch Project Parking**. With your group, adapt the code in **Scratch Project Parking** to write code for a game, **Park Your Car!**, with the following guidelines. You may use the Parking Grid image to create your backdrop or upload a parking grid of your own.

1. Use `when right arrow key pressed` block to move the sprite forward from one grid space to the next.
2. Use `when space key pressed` block to turn the sprite 90° in the direction of your choice.
3. Include two buildings in different spaces that the car must avoid when going to the parking spot.
4. When the sprite hits an obstacle or moves out of bounds, send a message to the player. You may use `touching edge ?` block to test if the moves are legitimate moves.
5. Use `touching color ?` block to test if the sprite properly arrives in the parking spot.

Have a group discussion answering the following questions before you start coding.

- How do you find the location on the grid to place your obstacles (the buildings)?
- How do you determine the value for 

block?
- What other questions do you have?

Be prepared to present the errors you made in the coding project and how you debugged and fixed the code.

Reflecting

What I Know About Testing and Debugging

- Work with a partner and write an entry in your math journal to describe how you will test your code in the future to make sure the code is correctly written.
- Discuss with your partner how you responded to these moments:
  - Frustration, knowing that there is an error in the code but not being able to fix it.
  - “Got it!” and you fixed the error.

Development Team

Authors

Dr. Marc Husband • Robbie Olivero • Sofia Saleem • Adrienne Scott • Gerrie Storr • Otto Wevers

Managing Editor – Dr. Marietta Bloch

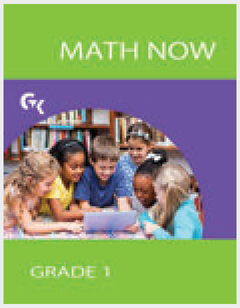
Mathematics Advisor – Dr. Marc Husband

Substantive Editor – Jodi Rauch

Copyeditor – Gerrie Storr

Learning Platform Design – Dr. Jovian Wat

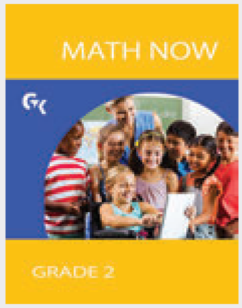
Illustrations – Pottery Chan



Student Resource
978-1-55317-481-3

Practising Fundamentals
978-1-55317-497-4

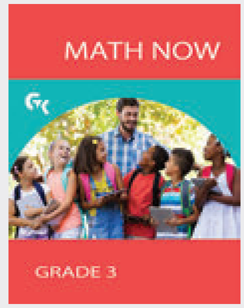
Teacher Resource
978-1-55317-489-9



Student Resource
978-1-55317-482-0

Practising Fundamentals
978-1-55317-498-1

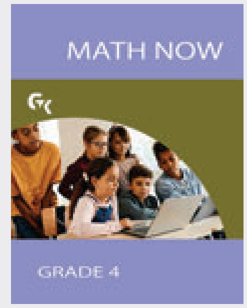
Teacher Resource
978-1-55317-490-5



Student Resource
978-1-55317-483-7

Practising Fundamentals
978-1-55317-499-8

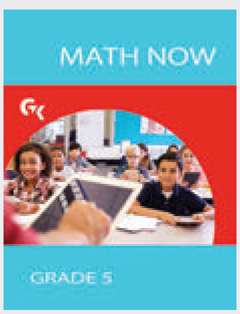
Teacher Resource
978-1-55317-491-2



Student Resource
978-1-55317-484-4

Practising Fundamentals
978-1-55317-500-1

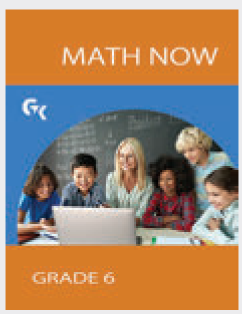
Teacher Resource
978-1-55317-492-9



Student Resource
978-1-55317-485-1

Practising Fundamentals
978-1-55317-501-8

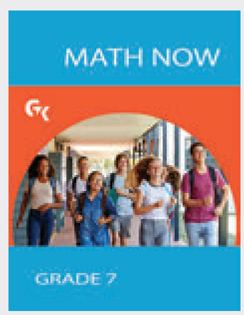
Teacher Resource
978-1-55317-493-6



Student Resource
978-1-55317-486-8

Practising Fundamentals
978-1-55317-502-5

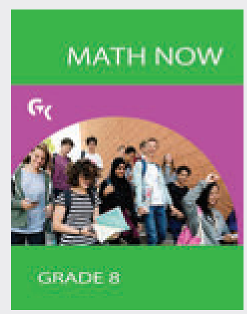
Teacher Resource
978-1-55317-494-3



Student Resource
978-1-55317-487-5

Practising Fundamentals
978-1-55317-503-2

Teacher Resource
978-1-55317-495-0



Student Resource
978-1-55317-488-2

Practising Fundamentals
978-1-55317-504-9

Teacher Resource
978-1-55317-496-7

Math Now Series Online (All Grades with Coding) 978-1-55317-298-7

Math Now Coding Online 978-1-55317-299-4

GTK Learning
https://learning.gtkpress.com
info@gtkpress.com
Phone (416) 385-1313
Fax (416) 385-1319
Unit 109, 18 Wynford Drive, Toronto, ON M3C 3S2