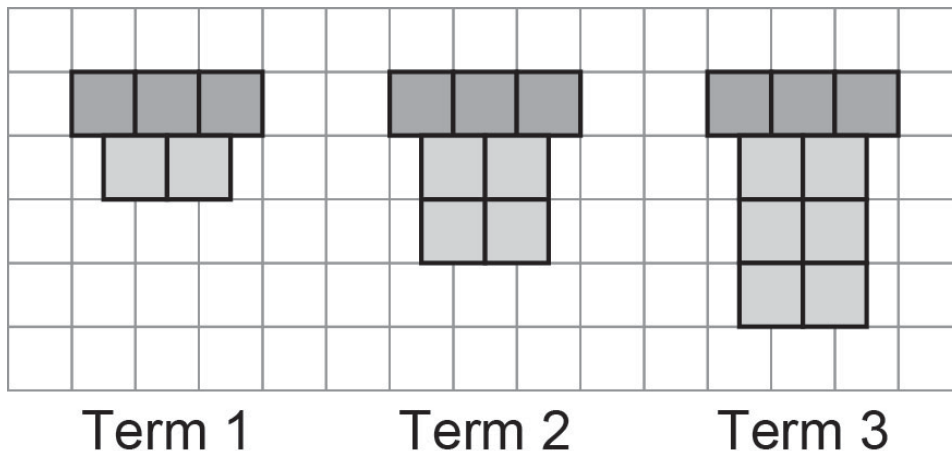


**Algebra**  
**Unit 1 Line Master 3f****Exploring Algebraic Expressions  
with Coding Answers**

There are many ways to represent algebraic expressions, including through words, tables of values, tiles, and graphs.

For example, the tiles below represent the first three terms of the expression  $2n + 3$ .



You can also represent algebraic expressions through coding.

1. Let's generate a table of values for any number of terms.

- a) Copy this exact code into a Python console, such as [Google Colab](https://colab.research.google.com/) or <https://cscircles.cemc.uwaterloo.ca/console/>.

```
for i in range (1,10):  
    termNumber = i  
    termValue = 2 * termNumber  
    print (termNumber, '\t', termValue)
```

**About the Code:** “for” indicates the start of a repeat. Everything that is indented below *for*, is included in the repeat. The first time through the repeat, *i* is 1, then 2, then 3, up to but not including 10.  
The **termNumber** variable is set to the value of *i*.

**Algebra**  
**Unit 1 Line Master 3g****Exploring Algebraic Expressions**  
**with Coding Answers (cont'd)**

When you execute the code by running it,

your output should look like this:

b) What is the numerical coefficient, or multiplier?

**The multiplier is 2.**

c) What algebraic expression does the output represent?

**The algebraic expression is  $2n$ .**

1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16
9	18

2. Alter the code so that it will output the numbers 1 through 100, **including** 100:

```
for i in range (1,101):  
    termNumber = i  
    termValue = 2 * termNumber  
    print (termNumber, '\t', termValue)
```

Describe the output generated with the altered code.

**The output displays termNumber 1 through 100 and the corresponding termValue, starting at 2 and ending at 200 for the 100th term number. The expression doubles the termNumber each time to obtain the termValue.**

**About the Code:**

\* means to multiply in coding.

- Each time through the loop, the values of the **termNumber** and **termValue** variables are updated and output.
- So, the first time through the loop, **termNumber** is 1 and **termValue** is  $2 * 1$ , which is 2.
- The second time through the loop, **termNumber** is 2 and **termValue** is  $2 * 2$ , which is 4.
- The third time through the loop, **termNumber** is 3 and **termValue** is  $2 * 3$ , which is 6.

And so on!

**Algebra**  
**Unit 1 Line Master 3h****Exploring Algebraic Expressions  
with Coding Answers (cont'd)**

3. Alter the code so that it will output a table of values with the first 50 terms of the expression  $3n$ .

```
for i in range (1,51):  
    termNumber = i  
    termValue = 3 * termNumber  
    print (termNumber, '\t', termValue)
```

- a) Describe the new output generated.

**The output displays termNumber 1 through 50 along with each termValue which will be the termNumber multiplied by 3. The first termValue is 3, and the last is 150.**

- b) What is the numerical coefficient, or multiplier for this expression?

**The numerical coefficient is 3.**

- c) What algebraic expression does the output represent?

**The algebraic expression is  $3n$ .**

4. Alter the code so that each of these outputs are generated:

a)

1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32
9	36
10	40

b)

1	6
2	12
3	18
4	24
5	30
6	36
7	42
8	48
9	54
10	60

c)

1	10
2	20
3	30
4	40
5	50

**Algebra**  
**Unit 1 Line Master 3i****Exploring Algebraic Expressions**  
**with Coding Answers (cont'd)****Sample Answers****a)**

```
for i in range (1,11):  
    termNumber = i  
    termValue = 4 * termNumber  
    print (termNumber, '\t', termValue)
```

**b)**

```
for i in range (1,11):  
    termNumber = i  
    termValue = 6 * termNumber  
    print (termNumber, '\t', termValue)
```

**c)**

```
for i in range (1,6):  
    termNumber = i  
    termValue = 10 * termNumber  
    print (termNumber, '\t', termValue)
```

5. Alter the code as shown below to display a table of values for the expression  $2n + 1$ , showing the output for **termNumbers** 0 to 15.

```
for i in range (1,16):  
    termNumber = i  
    termValue = 2 * termNumber + 1  
    print (termNumber, '\t', termValue)
```

**About the Code:** In this case, **termNumber** is starting at 0.

**Algebra**  
**Unit 1 Line Master 3j****Exploring Algebraic Expressions**  
**with Coding Answers (cont'd)**

Your output should look like this:

0	1
1	3
2	5
3	7
4	9
5	11
6	13
7	15
8	17
9	19
10	21
11	23
12	25
13	27
14	29
15	31

a) What is the numerical coefficient, or multiplier?

**The numerical coefficient is 2.**

b) What is the constant?

**The constant is 1.**

**Reflect and Connect**

6. a) How would you alter the code to represent the expression  $4n + 3$ , showing the output for **termNumbers** 0 through 100.

**Sample Answer**

```
for i in range (1,101):  
    termNumber = i  
    termValue = 4 * termNumber + 3  
    print (termNumber, '\t', termValue)
```

**Algebra**  
**Unit 1 Line Master 3k**

***Exploring Algebraic Expressions  
with Coding Answers (cont'd)***

b) Describe the output.

**The termNumber 0 through to 100 are displayed along with the corresponding termValue, starting at 3 and ending at 403.**

7. Alter the code to generate output using different expressions.  
Challenge your classmates by covering your code and having them try to determine what expression you used to produce the output generated.

**Answers will vary.**