Converting Algebraic Expressions to Programming Statements

Algebraic Expression	Operation	C++ Equivalent
6B	6 times B	6 *B
(3)(12)	3 times 12	3 * 12
4xy	4 times x times y	4 *x*y
y= 3x / 2		y = x / 2 * 3;
z= 3bc+4		z = 3 * b * c + 4;
a=3x+2 / 4a-1		a = (3 * x + 2) / (4 * a - 1)

Write code for the following:

The force of gravity between two masses is given by the formula

$$f = G\frac{M_1 M_2}{R^2}$$

where G is the gravitational constant, M1 and M2 are the masses of the two objects, and R is the distance between them. Translate this into a C++ expression, remembering to use good variable naming conventions.

You may assume that all variables have already been declared as doubles, that G has already been properly declared and initialized as a constant double, and that the values of all variables will be given in appropriate units - i.e. don't worry about units! grav_force = G_CONST * mass1 * mass2 / (separation * separation)

Write C++ expressions for the following algebraic expressions:

1. a=2b+4c
2. 3X + Y
3. $y=x^3$
4. $A^2 + 2B + C$
7. $g=\frac{x+2}{z^2}$
6. $g=\frac{x^2}{z^2}$
8. $\frac{\left(\frac{A^2+2B+C}{D}\right)}{XY}$
9. $\left(\frac{A+B}{C-D}\right) x \left(\frac{X}{Y}\right)$
10. **A PROGRAMMING CHALLENGE (EXTRA CREDIT):** Write a C++ program that computes the mean and standard deviation of a set of six integer values (data points). The mean is the sum of the six values divided by 6, and the formula for the standard deviation is:

$$M = \frac{\Sigma(X)}{N}$$

$$S^2 = \frac{\Sigma(X-M)^2}{n-1}$$

See next page for a brief explanation of these two formulas use in most statistical applications. The Programming challenge adapted from the Dale & Weems text Chapter 3 – Programming Problems

Calculating the Mean and Standard Deviation

Many of the questions from customer satisfaction or similar surveys include rating scales. This will require calculating means and standard deviations for data analysis. This can be done using popular spreadsheet software, such as Microsoft Excel®, or even online calculators. If neither of these is readily available, both the mean and standard deviation of a data set can be calculated using arithmetic formulas. Following are brief descriptions of the mean and standard deviation with examples of how to calculate each.

The Mean: For a data set, the mean is the sum of the observations divided by the number of observations. It identifies the central location of the data, sometimes referred to in English as the average. The mean is calculated using the following formula.

$$M = \frac{\Sigma(X)}{N}$$

Where Σ = Sum of X = Individual data points N = Sample size (number of data points)

Example: To find the mean of the following data set: 3, 2, 4, 1, 4, 4.

$$M = \frac{3+2+4+1+4+4}{6} = \frac{18}{6} = 3$$

The Standard Deviation: The standard deviation is the most common measure of variability, measuring the spread of the data set and the relationship of the mean to the rest of the data. If the data points are close to the mean, indicating that the responses are fairly uniform, then the standard deviation will be small. Conversely, if many data points are far from the mean, indicating that there is a wide variance in the responses, then the standard deviation will be large. If all the data values are equal, then the standard deviation will be zero. The standard deviation is calculated using the following formula.

$$S^2 = \frac{\Sigma(X-M)^2}{n-1}$$

Where Σ = Sum of X = Individual score M = Mean of all scores N = Sample size (number of scores)

Example: To find the Standard deviation of the data set: 3, 2, 4, 1, 4, 4.

Step 1: Calculate the mean and deviation.

X	M	(X-M)	$(X-M)^2$
3	3	0	0
2	3	-1	1
4	3	1	1
1	3	-2	4
4	3	1	1
4	3	1	1

Step 2: Using the deviation, calculate the standard deviation

$$S^2 = \frac{(0+1+1+4+1+1)}{(6-1)} = \frac{8}{5} = 1.6$$

$$S = 1.265$$

What to Infer from the Mean and Standard Deviation

If the data points are close to the mean, indicating that the responses are fairly uniform, then the standard deviation will be small. Conversely, if many data points are far from the mean, indicating that there is a wide variance in the responses, then the standard deviation will be large. However, the standard deviation alone is not particularly useful without a context within which one can determine meaning.

A standard deviation of 1.265 with a mean of 3, as calculated in our example, is much different than a standard deviation of 1.265 with a mean of 12. By calculating how the standard deviation relates to the mean, otherwise known as the coefficient of variation (CV), you will have a more uniform method of determining the relevance of the standard deviation and what it indicates about the responses of your sample. The closer the CV is to 0, the greater the uniformity of data. The closer the CV is to 1, the greater the variability of the data.

$$\mathrm{CV} = \frac{S}{M}$$

Using our example of a standard deviation of 1.265 and a mean of 3, you will see that the coefficient of variation is rather large, indicating that the data has a great deal of variability with respect to the mean and there is not general consensus among the sample.

$$\mathrm{CV} = \frac{S}{M} = \frac{1.265}{3} = .42$$

Using the example of a standard deviation of 1.265 and a mean of 12, you will see that the coefficient of variation is rather small, indicating that the data has a greater deal of uniformity with respect to the mean and there is a general consensus among the sample.

$$\mathrm{CV} = \frac{S}{M} = \frac{1.265}{12} = .11$$

ALGORITHM WORKBENCH EXERCISE#2 – WRITING NUMERIC EXPRESSIONS

EXTRA CREDIT (10pts) - DUE WEDNESDAY February 20, 2020

Write an interactive C++ program that outputs the mean and standard deviation of a set of ten integer values (aka. data points to represent student exam scores) that are input by the user. The mean is the sum of the ten values divided by 10, and the formulas to compute the mean and standard deviation is as follows*:

$$\text{MEAN} = \frac{\Sigma(X)}{N}$$

$$S^2 = \frac{\Sigma(X-M)^2}{n-1}$$

For a brief explanation of these two formulas used in most statistical applications please see pg. 2 of this handout titled Coding_expression.pdf.

Although the individual values (i.e., scores) are integers, the results are floating-point values. Provide appropriate prompts for user input of ten integer values. Be sure to use proper program formatting and appropriate comments in your code. The output should be labeled clearly and formatted neatly.

This short algorithm workbench exercise will be accessible in your class CANVAS module for weeks 3-4 shortly. Please use the file upload option to submit your completed source code.

PLEASE NOTE: Since this short programming exercise is an "optional" i.e., not a required assignment, the 10 extra credit points will be added to your final point tally at the end of the Spring 2020 session.

Thanks, sujan!

Sujan Sarkar, Instructor
Santa Rosa Junior College
Computer Studies Department
Maggini Hall, 524-1873
Email: ssarkar@santarosa.edu

*If the formulas do not display correctly in the email please check the handout or google for the statistical formulas, "mean" and "standard deviation" .

```
// Program precedence.cpp  demonstrates the precedence of operators
#include <iostream>
#include<cmath>
#include<cstdlib>
using namespace std;

int main ()
{
   cout << 4 + 3 * 5  << endl;
   cout << (4 + 3) * 5  << endl;
   cout << 4 * 5 % 3 + 2  << endl;
   cout << (4 * (5 / 3) + 2)  << endl;

   int y,x=3;
 //  y=x*x*x;
   y=int(pow(x,3.0));
   cout<<y<<endl;
   system("PAUSE");
   return 0;
}
```

```
// Program expressions.cpp  demonstrates arithmetic expressions as well as prestandard type casting
#include <iostream>
#include <iomanip>    // For setw() and setprecision()

using namespace std;
int main()
{

   int i = 4;
   int j = 17;
   float z = 2.6;

   cout << fixed << showpoint;            // Set up floating pt. output format

   cout <<"Variables declared and assigned below"<<endl;
   cout << "int i = 4;";
   cout << "int j = 17;";
   cout <<"float z = 2.6;"<<endl<<endl;

   cout<<"Testing various arithmetic expressions"<<endl;
   cout << "i/ float (j)= "<< i/ float (j) <<endl;
   cout << "1.0 / i +2 = "<< 1.0 / i +2 <<endl;
   cout << "z * j = "<< z * j <<endl;
   cout << "i + j % i = "<< i + j % i <<endl;
   cout << "(1/2)*i = "<< (1/2)*i <<endl;
   cout << "2*i+j-1 = "<< 2*i+j-1 <<endl;
   cout << "j/2 = "<< j/2 <<endl;
   cout << "2*3-1%3 = "<< 2*3-1%3 <<endl;
   cout << "i % j/i = "<< i % j/i <<endl;
   cout << "int(z + 0.5) = "<< int(z + 0.5) <<endl;
   return 0;
}
```

```
// Program modulus.cpp converts seconds to minutes and seconds.
// note the use of the modulus operator
#include <iostream>
#include<cstdlib>
using namespace std;

int main()
{
        // The total seconds is 125.
        int totalSeconds = 125;

        // Variables for minutes and seconds
        int minutes, seconds;

        // Get the number of minutes.
        minutes = totalSeconds / 60;

        // Get the remaining seconds.
        seconds = totalSeconds % 60;

        // Display the results.
        cout << totalSeconds << " is equivalent to:\n";
        cout << "Minutes: " << minutes << endl;
        cout << "Seconds: " << seconds << endl;
        return 0;
}
```

Sample programming challenges work with Numeric Expressions

PC3. Test Average

Write a program that asks for five test scores. The program should calculate the average test score and display it. The number displayed should be formatted in fixed-point notation, with one decimal point of precision.

PC12. Celsius to Fahrenheit

Write a program that converts Celsius temperatures to Fahrenheit temperatures. The formula is

$$F = \frac{9}{5} C + 32$$

F is the Fahrenheit temperature and C is the Celsius temperature. The program should prompt the user to input a Celsius temperature and should display the corresponding Fahrenheit temperature.

PC18. Interest Earned

Assuming there are no deposits other than the original investment, the balance in a savings account after one year may be calculated as

$$Amount = Principal * (1 + \frac{Rate}{T})^{T}$$

Principal is the balance in the account
Rate is the annual interest rate,
T is the number of times the interest is compounded during a year (e.g., T is 4 if the interest is compounded quarterly).

Write a program that asks for the principal, the interest rate, and the number of times the interest is compounded. It should display a report similar to the following:

Interest Rate:	4.25%
Times Compounded:	12
Principal:	$ 1000.00
Interest:	$ 43.33
Final balance:	$ 1043.33

PC24. Planting Grapevines

A vineyard owner is planting several new rows of grapevines, and needs to know how many grapevines to plant in each row. She has determined that after measuring the length of a future row, she can use the following formula to calculate the number of vines that will fit in the row, along with the trellis end-post assemblies that will need to be constructed at each end of the row:

$$V = \frac{R - 2E}{S}$$

The terms in the formula are:
V is the number of grapevines that will fit in the row.
R is the length of the row, in feet.
E is the amount of space, in feet, used by an end-post assembly.
S is the space between vines, in feet.
Write a program that makes the calculation for the vineyard owner. The program should ask the user to input the following:
The length of the row, in feet
The amount of space used by an end-post assembly, in feet
The amount of space between the vines, in feet
Once the input data has been entered, the program should calculate and display the number of grapevines that will fit in the row.

```
// Program 3-34 demonstrates using the C++ time function
// to provide a "seed" for the random number generator.
#include <iostream>
#include <cstdlib>      // Header file needed to use srand and rand
#include <ctime>        // Header file needed to use time
using namespace std;

int main()
{
        unsigned seed;        // Random generator seed
        // Use the time function to get a "seed" value for srand
        seed = time(0);
        srand(seed);

        // Now generate and print three random numbers
        cout << rand() << "     " ;
        cout << rand() << "     " ;
        cout << rand() << endl;
        return 0;
}
```

```
// This program 3-35 demonstrates random numbers, providing
// a "seed" calling the time function, and generating random numbers between 1 and 6.
#include <iostream>
#include <cstdlib>          // Needed to use rand() and srand()
#include <ctime>            // Needed to "seed" the random number generator
using namespace std;

int main()
{
        int num1, num2, num3;  // These hold the 3 random numbers
        unsigned seed;          // Seed for the random number generator
        int max_value = 6;
        seed = time(0);         // Assigns a system-generated seed

        // Set the random generator seed before calling rand()
        srand(seed);

        // Now generate random numbers to an integer between 1 and max_value and print three random numbers
        num1 = 1+ rand()% max_value;
        num2 = 1+ rand()% max_value;
        num3 = 1+ rand()% max_value;
        cout << num1 << "  " << num2 << "  "<<num3<<endl;
        return 0;
}
```