2.5 Arithmetic expressions (int)

Below is a simple program that includes an expression involving integers.

Figure 2.5.1: Expressions examples: Leasing cost.

```
#include <iostream>
using namespace std;
/* Computes the total cost of leasing a car given the down payment,
  monthly rate, and number of months
int main() {
   int downPayment;
   int paymentPerMonth;
   int numMonths;
   int totalCost; // Computed total cost to be output
   cout << "Enter down payment: ";</pre>
   cin >> downPayment;
   cout << "Enter monthly payment: ";</pre>
   cin >> paymentPerMonth;
   cout << "Enter number of months: ";</pre>
   cin >> numMonths;
   totalCost = downPayment + (paymentPerMonth * numMonths);
   cout << "Total cost: " << totalCost << endl;</pre>
   return 0;
```

Enter down payment: 500 Enter monthly payment: 300 Enter number of months: 60

Total cost: 18500

Feedback?

PARTICIPATION ACTIVITY

 $2.5.1: Simple \ program \ with \ an \ arithmetic \ expression.$



Consider the example above.

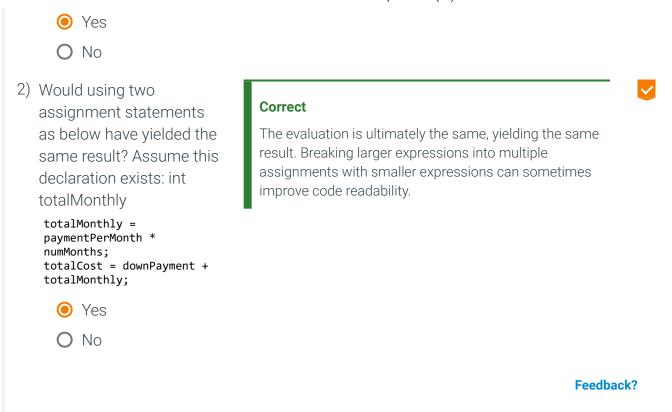
 Would removing the parentheses as below have yielded the same result?

downPayment +
paymentPerMonth *
numMonths

Correct

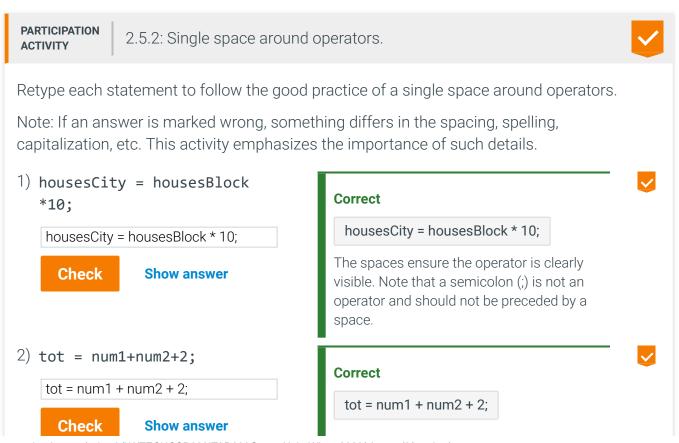
The * would still be evaluated first, because * has higher precedence than +. But using parentheses makes the programmer's intent more clear to people reading the program.



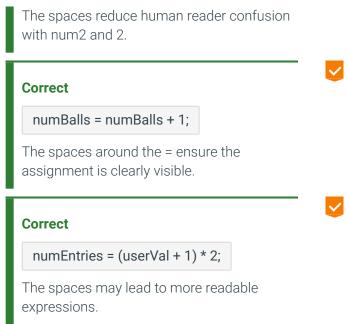


Style: Single space around operators

A good practice is to include a single space around operators for readability, as in numltems + 2, rather than numltems+2. An exception is minus used as negative, as in: xCoord = -yCoord. Minus (-) used as negative is known as **unary minus**.



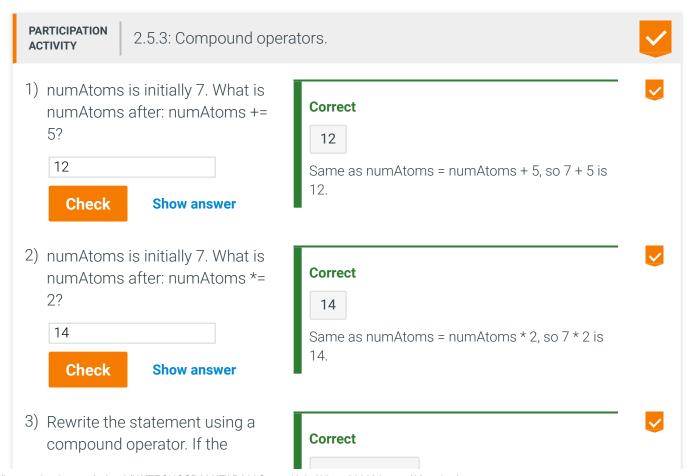
2.5. Arithmetic expressions (int)



Feedback?

Compound operators

Special operators called **compound operators** provide a shorthand way to update a variable, such as userAge **+=** 1 being shorthand for userAge = userAge + 1. Other compound operators include **-=**, ***=**, **/=**, and **%=**.



carCount /= 2;

statement can't be rewritten using a compound operator, type: Not possible carCount = carCount / 2;

carCount /= 2;

Check

Show answer

4) Rewrite the statement using a compound operator. If the statement can't be rewritten using a compound operator, type: Not possible numltems = boxCount + 1;

Not possible

Check

Show answer

Correct

Not possible

A compound operator is shorthand for when a variable is being updated, not for a more general assignment. Both sides must thus have the same variable, but here the variables are different (numItems and boxCount).

carCount /= 2 is same as carCount = carCount / 2.

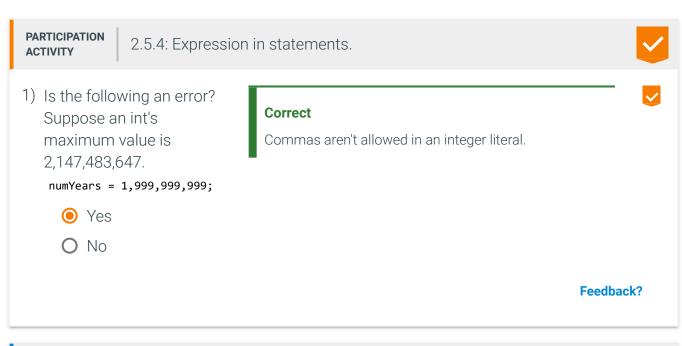
Feedback?

No commas allowed

CHALLENGE

ACTIVITY

Commas are not allowed in an integer literal. So 1,333,555 is written as 1333555.



2.5.1: Compute an expression.

Write a statement that assigns finalResult with the sum of num1 and num2, divided by 3. Ex: If num1 is 4 and num2 is 5, finalResult is 3.

```
1 #include <iostream>
   2 using namespace std;
   4 int main() {
         int num1;
         int num2;
   6
         int finalResult;
   8
   9
         cin >> num1;
   10
         cin >> num2;
   11
   12
         /* Your solution goes here */
   finalResult = (num1 + num2)/3;
   14
   15
         cout << "Final result: " << finalResult << endl;</pre>
   16
         return 0;
   17
   18 }
  Run
           All tests passed
Testing with 4, 5
            Your output
                            Final result: 3

✓ Testing with 4, 11

            Your output
                            Final result: 5
                                                                                  Feedback?
CHALLENGE
             2.5.2: Total cost.
ACTIVITY
A drink costs 2 dollars. A taco costs 4 dollars. Given the number of each, compute total
cost and assign totalCost with the result. Ex: 4 drinks and 6 tacos yields totalCost of
32.
   1 #include <iostream>
   2 using namespace std;
   4 int main() {
```

```
int numDrinks;
   6
        int numTacos;
        int totalCost;
   8
   9
        cin >> numDrinks;
  10
        cin >> numTacos;
  11
  12 /* Your solution goes here */
        totalCost = numDrinks* 2 + numTacos*4;
  13
  14
        cout << "Total cost: " << totalCost << endl;</pre>
  15
  16
  17
        return 0;
  18 }
          ✓ All tests passed
  Run

✓ Testing with 4 drinks and 6 tacos

           Your output
                           Total cost: 32

✓ Testing with 2 drinks and 8 tacos

           Your output
                           Total cost: 36
                                                                                Feedback?
```