



Students:  
Section 2.5 is a part of 2 assignments: **CSC108 CH02.1-2.10 C2A** ▾

Includes: CA

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See [this article](#) for more info.

Due: 02/04/2025, 11:59 PM EST

## 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: ";
    cin >> downPayment;

    cout << "Enter monthly payment: ";
    cin >> paymentPerMonth;

    cout << "Enter number of months: ";
    cin >> numMonths;

    totalCost = downPayment + (paymentPerMonth * numMonths);

    cout << "Total cost: " << totalCost << endl;

    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.

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

downPayment + paymentPerMonth \* numMonths;

- ☐ Yes  
☐ No

2) Would using two assignment statements as below have yielded the same result? Assume this declaration exists: int totalMonthly

totalMonthly = paymentPerMonth \* numMonths;  
totalCost = downPayment + totalMonthly;

- ☐ Yes  
☐ No

Feedback?

### Style: Single space around operators

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

### PARTICIPATION ACTIVITY

2.5.2: Single space around operators.



Retype each statement to follow the good practice of a single space around operators.

Note: If an answer is marked wrong, something differs in the spacing, spelling, capitalization, etc. This activity emphasizes the importance of such details.

1) `housesCity = housesBlock *10;`

**Check** **Show answer**

2) `tot = num1+num2+2;`

**Check** **Show answer**

3) `numBalls=numBalls+1;`

**Check** **Show answer**

4) `numEntries = (userVal+1)*2;`

**Check** **Show answer**

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 `%=`.

### PARTICIPATION ACTIVITY

2.5.3: Compound operators.



1) `numAtoms` is initially 7. What is `numAtoms` after: `numAtoms += 5`?

**Check** **Show answer**

2) `numAtoms` is initially 7. What is `numAtoms` after: `numAtoms *= 2`?

**Check** **Show answer**

3) Rewrite the statement using a compound operator. If the statement can't be rewritten using a compound operator, type: Not possible

`carCount = 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

`numItems = boxCount + 1;`

**Check** **Show answer**

Feedback?

### No commas allowed

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

### PARTICIPATION ACTIVITY

2.5.4: Expression in statements.



1) Is the following an error? Suppose an int's maximum value is 2,147,483,647.

```
int numYears;
numYears = 1,999,999,999;
```

- ☐ Yes  
☐ No

Feedback?

### CHALLENGE ACTIVITY

2.5.1: Calculate the values of the integer expressions.



620890 5010016 w32mq/7

**Start**

Type the program's output

```
#include <iostream>
using namespace std;

int main() {
    int x;
    int y;

    x = 4;
    y = x * 1;

    cout << y << endl;

    return 0;
}
```

4

1 2 3 4

**Check** **Next**

Feedback?

### CHALLENGE ACTIVITY

2.5.2: Arithmetic expressions (int).



620890 5010016 w32mq/7

**Start**

Integer variables `height` and `length` are read from input, representing the height and the length of a rectangle, respectively. Assign `recPerimeter` with the perimeter of the rectangle.

Ex: If the input is 20 11, then the output is:

Perimeter: 62

Note: The perimeter of a rectangle is calculated by multiplying the sum of the height and the length by 2.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int height;
6     int length;
7     int recPerimeter;
8
9     cin >> height;
10    cin >> length;
11
12    /* Your code goes here */
13
14    cout << "Perimeter: " << recPerimeter << endl;
15
16    return 0;
17 }
```

1 2

**Check** **Next level**

Feedback?

How was this section? **Provide section feedback**

Activity summary for assignment: **CSC108 CH02.1-2.10 C2A** ▾

41 / 41 points

Due: 02/04/2025, 11:59 PM EST

41 / 41 points submitted to

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See [this article](#) for more info.

BlackboardLearn

**Completion details** ▾