Arithmetic Operators


```
//*************************************************************************
// File name: 02Rectangle.cpp
// Author: Chadd Williams
// Date: 09/08/06
// Assignment: Lab02 Challenge
// Purpose: Calculate and display the area and perimeter of a rectangle based on the
//          length and width entered by a user.
//*************************************************************************

#include "stdafx.h"
#include <iostream>

using namespace std;

// The main function
int main()
{
        /*const*/ int LENGTH /* = 8*/;
        /*const*/ int WIDTH  /* = 3*/;

        int area;
        int perimeter;

        cout << "Please enter the width of the rectangle: ";
        cin >> WIDTH;
        cout << "Please enter the length of the rectangle: ";
        cin >> LENGTH;

        perimeter =  (2 * LENGTH) + (2 * WIDTH);

        area = LENGTH * WIDTH;

        cout << "The area of the rectangle is: " << area << ".\n" ;
        cout << "It was calculated with the formula: area = LENGTH * WIDTH."<<endl;
        cout << "The perimeter of the rectangle is: " << perimeter << "." ;
        cout << "\nIt was calculated with the formula: ";
```

Today

- Arithmetic Operators & Expressions
  - sections 2.15 & 3.2
  - Computation
  - Precedence
  - Algebra vs C++
  - Exponents

6.1 Practice

- Can you spot what is incorrect in the following program:

```
int main()
{
   const int pi = 3.14;
   double num;
   int i,j;

   num = e2;
   i = 4,000;
   ch = "b"; j = i;
   pi = 5;

   return 0;
}
```


Associativity

- The order in which an operator works with its operands
  - *left to right* or *right to left*

Operator	Associativity
(unary negation) –	Right to left
* / %	Left to right
+ -	Left to right

Grouping!

- To override precedence we use grouping symbols, ( )
  - `average = ( a + b +c ) / 3;`
- (3 + 12) * 2 - 3
- 4 + 17 % (3 + 9)
- 6 - 2 * 9 / ((3 * 4) – 9)
  - Work from the inside ( ) outward

Algebraic Expressions

- 4x + 16
  - This means: 4 * x + 16
  - In C++, we must always write the *

```
int result;
int sum;
result = 4sum + 8; // syntax error
result = 4 * sum + 8;
```


Exponents

- The exponent operator was missing from the list! x^2 y^n
- C++ does not provide an exponent operator as part of the language
- Use **pow()** in the **cmath** library

```
#include <cmath>
double area;
area = pow(4, 2); // area = 4^2
```


pow()

- **pow()** is not an operator
  - it is a *function*
  - like **main()**
  - **double pow(double x, double y)**
  - it takes as arguments two **doubles**
    - **x and y**
  - it produces a **double**

6.2 Practice

```
// Where are the errors?
#include <cmath>
double value, exp, result;
int number;
result = pow(value,exp);
14 = pow(value,2);
number = pow(value,exp);
result = 14 + pow(value,99);
```


Input

- Input operator (extraction operator): **>>**
- Gets input from some device/file
- Standard input (from keyboard): **cin**
- Whatever the user types in is stored in the variable to the right of the operator (the right operand)
- That variable must have already been declared
  - Given a data type and allocated space in memory
- When reading in the data typed by the user
  - Any spaces before the data item are skipped
  - Continues to read until the user hits return

Advanced Input & Output (3.1 & 3.8)

```
int value, value2;
cout << "Please enter an integer: ";
cin >> value;
cout << "Please enter two integers";
cout << " separated by some spaces: ";
cin >> value >> value2;
cout << value2 << value <<endl;
```

```
Please enter an integer: 6
Please enter two integers separated by some spaces: 9   42
42 9
```

Advanced Input

```
int value;
float dec;
char letter;
cout << "Please enter one integer, ";
cout << "one float and one character: ";
cin >> value >> dec >> letter;
cout << value << ' ' << dec << " " ;
cout << letter << endl;
```

```
Please enter one integer, one float and one character: 9  3.2  c
9 3.2 c
```


6.3 Practice

- What is the Output? Assume x = 2, y = 3

```
cout << x;
cout << x + x;
cout << "x=";
cout << x + y << " = " << y + x;
z = x + y;
cin >> x >> y;
// cout << "x + y = " << x + y;
cout << "\n";
```


Advanced Output

- How can we force output to look a particular way?
  - Precision of numbers
  - Spacing around output

```
Here are some floating point numbers:
       72.0
       72.00
       72.000
Here is a table of data:
         4   cat     15
       100     6    2.1
```


Precision of numbers

```
#include <iostream>
#include <iomanip> //New Library!
using namespace std;
int main()
{
  double number = 3.141592653589793;
  cout << number << endl; // default output
  cout << fixed << setprecision(1) << number << endl;
  cout << fixed << setprecision(2) << number << endl;
  cout << fixed << setprecision(3) << number << endl;
  cout << fixed << setprecision(4) << number << endl;
  return 0;
}
```

```
3.14159
3.1
3.14
3.142
3.1416
```

These numbers are *rounded*!

Explore on your own what happens if **`number`** is an integer.

Spacing around output

```
#include <iostream>
#include <iomanip> //New Library!
#include <string>
using namespace std;
int main()
{
  double number = 3.141592653589793;
  string name = "cs150";
  int integer = 42;
  cout << setw(6) << name << setw(6) << integer << endl;
  cout << setw(6) << fixed << setprecision(3) << number;
  cout << setw(4) << integer <<endl;
  return 0;
}
```

```
•cs150••••42
•3.142••42
```

A • represents a blank space

6.4 Practice

- Write a C++ program that allows the user the ability to enter their name and the number of nickels and pennies they have. You are then to print the number of dollars and change that corresponds to. The change should be in the form of nickels and pennies

Summary

- Today we have looked at:
  - Arithmetic Operators & Expressions
- Next time we will look at:
  - Typecasting
- Completed sections 3.2 & 3.8