CS161: Introduction to Computer Science I

Week 2 (b)

10/2022

Today in CS161

Operators and Expressions

Using the Math Library

Output Formatting

Programming Style

Arithmetic Operators and Expressions

Arithmetic operators:

- Example:
  - x = 11 % 3 //→ x is 2
  - x = 11 / 3 // → x is 3

Operator	Decription
+	Addition
-	Substration
*	Multiplication
/	Division
%	Modulo

As in most other languages, C++ allows you to form expressions using variables, constants, and the arithmetic operators.

Compound assignment

Expression	Equivalent to
result += 10;	result = result + 10;
result -=10;	result = result – 10;
result *= x+y;	result = result * (x+y);
result /= x+y;	result = result / (x+y);

Division with Whole Numbers

When you use the division operator / on two **integers**, the result is an **integer.**

```
int midterm, final;  //midterm = 9, final =6
float gpa;
…
gpa = (midterm + final) / 2;   //gpa = 7.0
gpa = (midterm + final) / 2.0; //gpa = 7.5
```

Relational and Equality Operators

Relational Operators:

Operator	Decription
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal

Equality Operators:

Operator	Decription
==	Equal
!=	Not equal

Using the Math Library

In C++ there is no operator that will raise some number by another

For example, for x^3 you can’t type:

- ○ x**3 ILLEGAL!
- ○ x^3 ILLEGAL!

Instead, you must use the **pow** function

```
float answer;
answer = pow(x, 3);
```

Using the Math Library

To use the power function, we must:

- include the math library:

```
#include <cmath>
```

- Use the function: **pow(x, y);**

→ x: base, y: exponent

x≠0, y=0 ➔ pow(x, y) = 1

x=0, y=0 ➔ pow(x, y) = 1

x=0, y<0 ➔ pow(x, y) = INF

Using the Math Library

The cmath library contains the definition of some mathematical functions:

Name	Decription	Example	Value
sqrt	Square root	sqrt(4.0);	2.0
pow	Powers	pow(2.0,3.0);	8.0
fabs	Absolute value	fabs(-7000);	7000
ceil	Round up	ceil(3.2);	4.0
floor	Round down	floor(3.2);	3.0

…

Exercise

□ Convert each of the following mathematic expressions to a C++ arithmetic expression.

a) $\sqrt{x+y}$

b) x^{y+7}

c) $|x-y|$

d) $\frac{-b+\sqrt{b^2-4ac}}{2a}$

Formatting for Numbers with a Decimal Point

Output of the following **cout** statement may vary:

```
cout << “The price is $ “ << price << endl;
```

- The price is $10.5;
- The price is $10.5000;
- The price is $10.50000e01;
- The price is $10.50;

Next, to Format our Output

We must learn about **precision**

By default, real numbers are displayed with no more than 6 digits, plus the decimal point

This means that 6 significant digits are displayed in addition to the decimal point and a sign if the number is negative

Default Precision -- Examples

```
float test;
cout << “Please enter a real number”;
cin >> test;
cout << test;
```

Input	Resulting Output
1.23456789	1.23457
10.23456789	10.2346
100.23456789	100.235
1000.23456789	1000.23
100000.23456789	100000

To Change Precision

```
float test;
cout << "Please enter a real number";
cout.precision(3); //3 instead of 6!!
cin >> test;          cout << test << endl;
```

Input	Resulting Output
1.23456789	1.23
10.23456789	10.2
100.23456789	100
10000.23456789	1e+04
	(Exponential notation)

Another way to do this...

```
#include <iomanip>
float test;
cout << “Please enter a real number”;
cin >> test;
cout << setprecision(3) << test << endl;
```

- setprecision is a manipulator
- To use it, we must include the **`iomanip`** header file
- There is no difference between

`cout.precision(3)` and **`cout <<setprecision(3)`**

What is “width”?

The width of a field can be set with:

```
cout.width(size);
```

If what you are displaying cannot fit, a larger width is used

- to prevent the loss of information

Important

- Width is only in effect for the next output

How does width work...

```
float test;
cout.precision(4);
cout.width(10);
cin >> test;
cout << test;
cout << endl <<test;
```

Input	Resulting Output
1.23456789	1.235 1.235

Another way to do this...

```
#include <iomanip>
float test;
cout.precision(4);
cin >> test;
cout << setw(10) << test;
cout << endl <<test;
```

Input	Resulting Output
1.23456789	1.235 1.235

Trailing Zeros

For real numbers, trailing zeros are discarded when displayed

Input	Resulting Output
`1.2300`	`1.23` (for a precision of 3 or greater)

To display trailing zeros we use:

```
cout.setf(ios::showpoint);
```

Displaying Trailing Zeros

```
float test;
cout.precision(4);
cout.setf(ios::showpoint);
cin >> test;
cout << test << endl;
cout.unsetf(ios::showpoint); //reset...
cout <<test;
```

```
Input                     Resulting Output

1.2300                    1.230
                               1.23
```

Displaying Dollars and Cents!

There is another meaning to precision...

- if we put in our programs:

```
cout.setf(ios::fixed,ios::floatfield);
```

- then, subsequent precision applies to the number of digits after the decimal point!

```
cout.precision(2);   cout << test;
```

Input	Resulting Output
1.2300	**1.23**
1.20	**1.20**

Program Style

The Style of your program is important because by doing it cleanly, you can create programs that are easier to read and correct

Style includes...

1. *indentation*
2. *grouping like elements*
3. *using blank lines*
4. *variables and program names*

Poor Program Style

```
#include <iostream>
using namespace std;
int main() { float c; float f; cout
   <<"Please enter”
  <<“ temperature in Celsius: " <<endl;
   cin >>c; f = (c * 9.0/5.0) + 32.0; cout
   <<c;
   cout <<" Celsius = " <<f; cout <<"
   Fahrenheit"; cout <<endl;  return 0;}
```

Better Program Style

```
#include <iostream>
using namespace std;
//This program converts temperatures……
int main()
{
     float celsius;  //temp in celsius
     float fahr;     //temp in Fahrs

     //Read in the temperature in celsius
     cout << "Enter temp in Celsius: ";
     cin >> celsius;

     //Convert celsius to fahrenheits
     fahr = (celsius * 9.0 / 5.0) + 32.0;

     //Print the results
     cout << celsius << " Celsius  =  " << fahr;
     cout << " Fahrenheit" << endl;

     return 0;
}
```

Tips for Better Program Style

Use meaningful names for variable names

Careful indenting and comments

End each program with an **endl**

Use blank lines.

Format for outputting a real number.

Library and Namespace

C++ comes with a number of standard libraries. These libraries place their definitions in a *namespace*, which is simply a name given to a collection of definitions.

Using library:

- **#include <Library_name>**
- **using namespace std;**

Standard namespace defining all the standard libraries we will be using

Exercise

It is said that a dog’s age is equivalent to 7 times of a human age.

- Example: A 5 year old dog is equivalent to a 35 year old human.

Write an algorithm to calculate human age of a dog; then translate that algorithm to a C++ program:

- Input: the dog’s age
- Output: human age of the dog