C/C++ Programming for Engineers: Floating Point Mathematics

John T. Bell

Department of Computer Science
University of Illinois, Chicago

Preview

Which statement best declares and initializes the double variable?

A. double currHumidity = 99%;
B. double currHumidity = 99.0;
C. double currHumidity = 99;
D. double currHumidity = ( double ) 99;
E. double currHumidity = 99 percent;
Recall C++ has five basic types of data:

- Integers – Whole numbers with no fractions.
- Floating Point – Numbers that may contain fractions, and have a larger range due to exponents.
- Characters – Small integer codes representing individual letters and other symbols.
- Strings – Collections of zero or more characters.
- Booleans – Variables holding ‘True’ or ‘False’.

Floating point #s come in a few flavors:

- float – The simplest (and least precise) of the floating point types. Saves space at the expense of accuracy.
- double – Twice as many bits as an ordinary float. Most common type for scientific and engineering programming. (Think 3.14159 vs. 3.14)
- long double – Twice as many bits as a double. Used when very large or precise numbers needed.
A Comparison of Floating Point Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Total bits</th>
<th>Exponent</th>
<th>Bits precision</th>
<th>Number of decimal digits</th>
<th>Accuracy near zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>float</td>
<td>32</td>
<td>8</td>
<td>24</td>
<td>~7.2</td>
<td>~10^-7</td>
</tr>
<tr>
<td>double</td>
<td>64</td>
<td>11</td>
<td>53</td>
<td>~15.9</td>
<td>~10^-15</td>
</tr>
<tr>
<td>x86 extended precision</td>
<td>80</td>
<td>15</td>
<td>64</td>
<td>~19.2</td>
<td>~10^-19</td>
</tr>
<tr>
<td>long double</td>
<td>128</td>
<td>15</td>
<td>113</td>
<td>~34.0</td>
<td>~10^-34</td>
</tr>
<tr>
<td>int</td>
<td>32</td>
<td>NA</td>
<td>32</td>
<td>~10</td>
<td>NA</td>
</tr>
</tbody>
</table>

(Most common values shown. Some implementations may vary.)

Representation of Floating Point Literals

- **Leading non-zeros**, containing a decimal point: 42.0 [±]1-9[0-9...].[0-9...][Ee[±]0-9...][FfLI]
- A leading zero must be followed by a point: 0.618 [±][0].[0-9...][Ee[±]0-9...][FfLI]
- Exponential notation: 6.02E23 = 6.02 * 10^23 [±]1-9[0-9...][Ee[±]0-9...][FfLI]
- A trailing F indicates floating point, L is long: [±]1-9[0-9...][Ff][L]
- Short version: 42 is an int, 42.0 is a double, 42f or 42.0f is float, and 42L or 42.0L is long double.
Automatic Conversion of Data Types

• Assignment converts to the stored type:
  – int n = 3.14; // stores an integer 3
  – double x = 42; // stores double precision 42.0

• Binary operators promote to match types:
  – 1 / 3 yields 0, using integer math
  – 1.0 / 3 yields 0.3333333, promoting 3 to a double
  – 1 / 3.0 yields 0.3333333, promoting 1 to a double
  – Temporary variables created and used as needed.

Specified Conversion of Data Types

• C++ has 4 different defined ways to type cast data from one type to another.

• For ordinary data types, static_cast is best:

• average = total / static_cast<double>( nScores );

• C Style type casts also work, with some weaknesses beyond the scope of this course:

  o average = total / ( double ) nScores;
C++ Math Library

- `#include <cmath>` to use the standard library of floating point math functions. (From C)
- [http://www.cplusplus.com/reference/cmath/](http://www.cplusplus.com/reference/cmath/) has a full list. (Linked under resources.)
- Example - Square Root:
  - double sqrt( double ); // Defined “prototype”
  - Takes a double argument and yields a double.
  - X = Y + sqrt( Z ) + Q;

More commonly used math functions

- sin, cos, tan, acos, etc. – Trigonometric
- exp – Exponential (base e)
- log – Logarithm (base e)
- log10 – Logarithm (base 10)
- pow( a, b ) – Raise a to the b power.
  - (If b is 2 or 3, use multiplication instead.)
- fabs – Floating point absolute value.
  - abs( ) also works in C++, but only applies to ints in C.
C++ Function Overloading

- C++ Allows for multiple functions with the same name to be distinguished by the type of data passed to them. This function overloading is mostly beyond the scope of CS109.
  - `acos(-1.0)` calls a function that takes a double and returns a double.
  - `acos(-1.0f)` passes a float and returns a float.
  - `acos(-1.0L)` passes and returns long doubles.
  - `acos(-1)` makes the computer guess which to use.

Which of the following data types is most appropriate for most scientific and engineering calculations?

A. char  
B. int  
C. float  
D. double  
E. long double