Operators, Modifiers, and Math Functions

Hours 8 and 9

- Objectives
  - Modifiers: signed, unsigned, long, and short
  - Logical operators
  - True and false
  - Bitwise operators
  - Conditional operator (?):
  - sizeof operator
  - Math library

Variable Modifiers

C/C++ keywords

- signed  little used- turn unsigned char into signed char
  (the “signedness” of char is machine and/or implementation defined)

- unsigned  often used- turns short, int, and long into
  unsigned (i.e., no negative values) type
  (doubles the magnitude; often used with bit-manipulation operators; unsigned arithmetic does not overflow)

- long  “long int” is usually just called “long”
  - some compilers accept “long double”

- short  “short int” is usually just called “short”
Logical Operators

Return true or false

- Left and right hand operands are boolean expressions
- && logical and
  - 0 < i && i < 10
- || logical or
  - i <= 0 || i >= 10
- ! logical not
  - ! done
  - ! (x > 0 && x < 10) ⇔ x <= 0 || x >= 10
- Can be used together
  - ! done && x > 0 || x == 100

<table>
<thead>
<tr>
<th>E₁</th>
<th>E₂</th>
<th>E₁ &amp;&amp; E₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

| E₁ | E₂ | E₁ || E₂ |
|----|----|-----|-----|
| f  | f  | f   | f   |
| f  | t  | f   | t   |
| t  | f  | t   | f   |
| t  | t  | t   | t   |

<table>
<thead>
<tr>
<th>E</th>
<th>!E</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>

True and False

An existential dilemma?

- False is 0
- Any value except 0 is true
- The result of a “boolean” expression is either 1 or 0
- C++ defines type bool with possible values: true & false
  - Still implemented as 0 and not-0

    ```c
    if (n % 2)
      printf("n is odd\n");
    else
      printf("n is even\n");
    if (strcmp(s1, s2))
      printf("s1 & s2 differ\n");
    else
      printf("s1 & s2 are equal\n");
    ```
Operator Examples

More unusual C operators

- ?:  (conditional operator)
  - (exp1) ? (exp2) : (exp3)
  - if exp1 is non-zero, the value of the conditional expression is exp2
  - if exp1 is zero, the value of the conditional expression is exp3
  - expressions are placed in parentheses if more than a constant or variable
  - `max = (x > y) ? x : y;`
  - `count = (count == max) ? 0 : (count + 1);`
  - `index = (index+1 == size) ? 0 : ++index;`

- `sizeof`
  - returns the size, measured in bytes, of variables or data types
  - evaluated by the compiler (i.e., does not generate runtime code)
  - `sizeof(int) /* parentheses are required with data types */`
  - `sizeof x /* parentheses are optional with variable names */`
  - `sizeof(x)`

Bit Operations

bitwise operations

- `<<` (left shift)  `11001100 << 2 = 00110000`
- `>>` (right shift)
  - `11001100 >> 2 = 00110011` unsigned or w/o sign extend
  - `11001100 >> 2 = 11110011` signed and sign extend
  - `00110011 >> 2 = 00001100` signed, unsigned, or sign extend
- `&` (bitwise and)  `1100 & 1001 = 1000`
- `|` (bitwise or)  `1100 | 1001 = 1101`
- `^` (exclusive or)  `1100 ^ 1001 = 0101`
- `~` (complement)  `~1100 = 0011`
Math Library Functions

Available in C and C++

Common Math Functions
- double sin(double x);
- double cos(double x);
- double tan(double x);
- double asin(double x);
- double acos(double x);
- double atan(double x);
- double atan2(double y, double x);
- \( \tan^{-1}(y/x) \)
- double exp(double x);
- \( e^x \)
- double log(double x);
- double log10(double x);
- double pow(double x, double y);
- \( x^y \)

Using the math library
- #include <math.h>
- link with Unix math library -lm (bigger than Windows/DOS)
- included in Windows runtime
- all angles are in radians

Math library also includes
- Hyperbolic functions
- Bessel functions
- Gamma functions
- Absolute value functions
- floor and ceiling
- etc.