More Arrays

Last Time
- We
  - Learned how to pass arrays to functions
- Today we will
  - Revisit random number generation
  - Start talking about sorting arrays

Random Number Generation Revisited
- Remember, the library `<cstdlib>` contains a function for generating random numbers
- The statement used to produce integers in the range 0 - 5 is
  - `int x = rand() % 6;
- To simulate the role of a dice we would use the statement
  - `int x = 1 + rand() % 6`

Random Number Generation
- 21.1: Write a program that will simulate the roll of a dice 6000 times and show the frequency in which each side appeared

<table>
<thead>
<tr>
<th>Face</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1003</td>
</tr>
<tr>
<td>2</td>
<td>1017</td>
</tr>
<tr>
<td>3</td>
<td>983</td>
</tr>
<tr>
<td>4</td>
<td>994</td>
</tr>
<tr>
<td>5</td>
<td>1004</td>
</tr>
<tr>
<td>6</td>
<td>999</td>
</tr>
</tbody>
</table>

Sorting Arrays
- Bubble sort
- Not suitable for large arrays
- Smaller values gradually bubble their way upward to the top of the array

Bubble Sort
```c
for( int pass = 0; pass < size - 1; pass ++ )
    for( int j = 0; j < size - 1 - pass; j++ )
        if( a[j] > a[j+1] )
            { 
                hold = a[j]; 
                a[j] = a[j+1]; 
                a[j+1] = hold; 
            }
```
Summary

- In today’s lecture we covered
  - Random numbers
  - Sorting arrays (bubble sort)

- Readings
  - P. 276 - p. 278: Bubble sort
  - P. 262 - p. 264: Histograms
  - P. 182 - p. 188: Random number generation