Review from last Thursday’s Class and from Lab

- Strings: subscripting and type declarations
- C++ basics: Loops and loop invariants, conditionals, assignments and lvalues, operators, integral types.
- Problem solving: two methods of thinking about output along a diagonal.

Today’s Class

Koenig & Moo: Chapter 3

- vector container class,
- sort function,
- C++ and standard library:
  - double,
  - reading a list of values of unknown length,
  - i/o manipulation

For several of these items, the text provides a more thorough description than we will.

- Algorithms and problem solving: computing statistics

Problem: Grade Statistics

- Read an unknown number of grades.
- Compute:
  - Mean (average)
  - Standard deviation
  - Median (middle value)
  - Mode (most frequent)
Reading and Averaging

A simple program, called \texttt{average.cpp}, to read and average the grades is attached. It has just two new features:

- reading integers in a while loop,
- manipulating the precision of the output stream.

Standard Deviation

Computing the standard deviation involves a bit more:

- Definition: if \( a_0, a_1, a_2, \ldots, a_{n-1} \) is a sequence of \( n \) values, and \( \mu \) is the average of these values, then the standard deviation is

\[
\left[ \frac{\sum_{i=0}^{n-1} (a_i - \mu)^2}{n - 1} \right]^{\frac{1}{2}}
\]

- Computing this equation requires two passes through the values:
  - Once to compute the average
  - A second time to compute the standard deviation
- Thus we need a way to store the values...

Vectors

- Standard library “container class” to hold sequences.
- It acts like a dynamically-sized, one-dimensional array.
- Capabilities:
  - Holds objects of any type
  - Starts empty unless otherwise specified
  - Any number of objects may be added to the end — there is no limit on size.
  - It can be treated like an ordinary array using the subscripting operator, but one must be certain the subscripts are within the correct bounds.
Computing the Standard Deviation

The program to compute the standard deviation is attached (average_and_deviation.cpp). Our discussion will focus on the use of vectors and the computation of the standard deviation.

- **Header files:**
  
  ```cpp
  #include <vector>
  #include <cmath>
  ```

  includes the C standard math library prototypes and declarations so that we can use the `sqrt` function.

- **The following creates an empty vector of integers:**

  ```cpp
  vector<int> scores;
  ```

Vectors are an example of a *templated container class*.

- The angle brackets `< >` are used to specify the type of object (the “template type”) that will be stored in the vector.

- **`push_back`** is a vector function to append a value to the end of the vector, increasing its size by one.
  
  - There is NO corresponding `push_front` operation for vectors.

- **`size`** is a function defined by the vector type (the vector class) that returns the number of items stored in the vector.

- After vectors are initialized and filled in, they may be treated *just like ordinary arrays*.
  
  - The line
    ```cpp
    double error = scores[i] - average;
    ```
    illustrates use of subscripting operations on vectors.
  
  - We could also write statements like
    ```cpp
    scores[4] = 100;
    ```
to change a score.

- It is the job of the programmer to ensure that any subscript value $i$ that is used is legal — at least 0 and less than $\text{scores.size()}$.

- The variables $\text{sum_sq_error}$ and $\text{error}$ are of type $\text{double}$, which is similar to $\text{float}$, but more accurate.

**Median**

- Intuitively, a median value of a sequence is a value that is less than half of the values in the sequence, and greater than half of the values in the sequence.

- More technically, if $a_0, a_1, a_2, \ldots, a_{n-1}$ is a sequence of $n$ values AND if the sequence is sorted such that $a_0 \leq a_1 \leq a_2 \leq \cdots \leq a_{n-1}$ then the median is

$$\begin{cases} 
\frac{a_{(n-1)/2}}{} & \text{if } n \text{ is odd} \\
\frac{a_{n/2-1} + a_{n/2}}{2} & \text{if } n \text{ is even}
\end{cases}$$

- Sorting is therefore the key to finding the median.

**Computing Averages and Medians**

A program, `average_and_median.cpp`, that uses vectors to compute the average, standard deviation, and median of a set of grades is attached to these notes. Here are features of the code we will discuss in detail:

- $\text{scores.begin()}$ returns a value that denotes the beginning of (first entry in) the vector; $\text{scores.end()}$ returns a value that denotes one position past the end of the vector.

- The generic function $\text{sort}$ is used to sort the interval from $\text{scores.begin()}$ up to (but not including) $\text{scores.end()}$.

- The header file $\text{algorithm}$ contains the function prototypes for “generic functions” in the standard library such as $\text{sort}$.

  - We will study and use a number of generic functions. They are efficient and flexible.

- The calculation of the median is straightforward after sorting.
Computing the Mode

- Recall that the mode is the most frequently occurring value.
- Sorting has placed repeated scores adjacent to each other in the vector, e.g.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>43</td>
<td>46</td>
<td>52</td>
<td>56</td>
<td>56</td>
<td>58</td>
<td>62</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>74</td>
<td>78</td>
</tr>
</tbody>
</table>

(Note how we draw the vector like we would an array.)
- We can exploit this by counting repeated occurrences as we step through the vector, one location at a time:
  - When the current value (at location \( i \)) is the same as the previous value (at location \( i - 1 \)), we just need to increment a counter variable.
  - When the current value (at location \( i \)) differs from the previous value (at location \( i - 1 \)), we’ve reached the end of a sequence of one or more instances of the same value (ending at location \( i - 1 \)). We need to
    * Check to see if a new mode has been found, recording the value and its count if it has.
    * Reset the count to 1 (not 0).
- At the end, we need to check to see if the last value was the mode.

Final Version of the Program

The code that computes the average, standard deviation, median and mode is in `average_median_mode.cpp`

- The algorithm as described above is implemented in the last part of the program.
- Of special note, look at and think about the loop invariant for computing the mode.

Looking Ahead to Chapter 4

There’s a lot of material, but we will not cover all of it.

- You are responsible for
– Functions and parameter passing; this should be review
– Separating program code into different files
– Structs

• You need to look at and learn how to use output formatting.
• You do NOT need to learn about exceptions and exception handling!