PROGRAMMING PROJECTS

For more in-depth projects, please see www.cs.colorado.edu/~main/projects/

1 Specify, design, and implement a class that can be used in a program that simulates a combination lock. The lock has a circular knob, with the numbers 0 through 39 marked on the edge, and it has a three-number combination, which we'll call *x, y, z.* To open the lock, you must turn the knob clockwise at least one entire revolution, stopping with *x* at the top; then turn the knob counterclockwise, stopping the *second* time that *y* appears at the top; finally turn the knob clockwise again, stopping the next time that *z* appears at the top. At this point, you may open the lock.

Your `lock` class should have a constructor that initializes the three-number combination (use 0, 0, 0 for default arguments). Also provide member functions:

(a) to alter the lock's combination to a new three-number combination
(b) to turn the knob in a given direction until a specified number appears at the top
(c) to close the lock
(d) to attempt to open the lock
(e) to inquire about the status of the lock (open or shut)
(f) to tell you what number is currently at the top

2 Specify, design, and implement a class called `statistician`. After a statistician is initialized, it can be given a sequence of double numbers. Each number in the sequence is given to the statistician by activating a member function called `next_number`. For example, we can declare a statistician called `s`, and then give it the sequence of numbers 1.1, –2.4, 0.8 as shown here:

```
statistician s;
s.next_number(1.1);
s.next_number(-2.4);
s.next_number(0.8);
```

After a sequence has been given to a statistician, there are various member functions to obtain information about the sequence. Include member functions that will provide the length of the sequence, the last number of the sequence, the sum of all the numbers in the sequence, the arithmetic mean of the numbers (i.e., the sum of the numbers divided by the length of the sequence), the smallest number in the sequence, and the largest number in the sequence. Notice that the length and sum functions can be called at any time, even if there are no numbers in the sequence. In this case of an "empty" sequence, both length and sum will be zero. But the other member functions all have a precondition requiring that the sequence is non-empty.

You should also provide a member function that erases the sequence (so that the statistician can start afresh with a new sequence).

Notes: Do not try to store the entire sequence (because you don't know how long this sequence will be). Instead, just store the necessary information about the sequence: What is the sequence length? What is the sum of the numbers in the sequence? What are the last, smallest, and largest numbers? Each of these pieces of information can be stored in a private member variable that is updated whenever `next_number` is activated.

3 Overload the + operator to allow you to add two statisticians from the previous project. If `s1` and `s2` are two statisticians, then the result of `s1 + s2` should be a new statistician that behaves as if it had all of the numbers of `s1` followed by all of the numbers of `s2`.

4 Specify, design, and implement a class for a card in a deck of playing cards. The object should contain methods for setting and retrieving the suit and rank of a card.

5 Specify, design, and implement a class that can be used to keep track of the position of a point in three-dimensional space. For example, consider the point drawn at the top of the next column. The point shown there has three coordinates: