Section Solutions 1

Problem 1: Random Shuffling

Here is one possible solution:

```
string randomShuffle(string input) {
      /* Base case: There is only one possible permutation of a string
       * with no characters in it.
       */
      if (input == "") {
            return input;
      } else {
            /* Choose a random index in the string. */
            int i = randomInteger(0, input.length() - 1);

            /* Pull that character to the front, then permute the rest of
             * the string.
             */
            return input[i] +
                   randomShuffle(input.substr(0, i) + input.substr(i + 1));
      }
}
```

This function is based on the recursive observation that there is only one possible random shuffle of the empty string (namely, itself), and then using the algorithm specified in the handout for the recursive step.

Problem 2: Computing Averages

```
void averageValueInFile(string filename, double& result) {
      /* Track how many total numbers have been read so far. */
      int numValues = 0;

      /* Clear the result; we'll add up all the values we find. */
      result = 0;

      /* Open the file for reading.  Note the use of .c_str() here. */
      ifstream input(filename.c_str());

      /* Use the canonical "read all the values in a file" loop to read
       * all the real numbers.
       */
      double val;
      while (input >> val) {
            numValues ++;
            result += val;
      }

      result /= numValues;
}
```

Problem 3: Haiku Detection

```
int syllablesIn(string word);
bool isHaiku(string line1, string line2, string line3);
int syllablesInLine(string line);

int main() {
      string line1 = getLine("Enter the first line: ");
      string line2 = getLine("Now, enter the second line: ");
      string line3 = getLine("Enter the third line: ");

      /* Given these three lines,
       * check whether they're a haiku,
       * then show the result.
       */
      if (isHaiku(line1, line2, line3)) {
            cout << "The text you entered" << endl;
            cout << "Goes 5 - 7 - 5, so it" << endl;
            cout << "is a haiku.  Yay!" << endl;
      } else {
            cout << "Though you have tried hard," << endl;
            cout << "The three lines you entered are" << endl;
            cout << "Not a haiku.  Awww." << endl;
      }

      return 0;
}

/* Given a poem
 * of three lines, returns whether
 * it is a haiku.
 */
bool isHaiku(string line1, string line2, string line3) {
      return syllablesInLine(line1) == 5 &&
             syllablesInLine(line2) == 7 &&
             syllablesInLine(line3) == 5;
}

/* Counts the number of
 * syllables in a line of
 * text, then returns it.
 */
int syllablesInLine(string text) {
      /* To split apart the
       * text, make a TokenScanner
       * and configure it.
       */
      TokenScanner scanner(text);
      scanner.ignoreWhitespace();

      int numSyllables = 0;
      while (scanner.hasMoreTokens()) {
            /* If this token is
             * a word, count its syllables
             * and update total.
             */
            string token = scanner.nextToken();
            if (scanner.getTokenType(token) == WORD) {
                  numSyllables += syllablesIn(token);
            }
      }

      return numSyllables;
}
```