

### Write a Program: DNA Sequence Revisited

Use Assignment 5c as your starting point.

Every DNA sequence consists of four nucleotides: Adenine, Thymine, Cytosine, and Guanine, referred to by the first letters of their chemical names (A, T, C, and G). I have provided an entire DNA sequence in the file **dnaSequence.txt**.

Your goal is to provide a report of the number of each nucleotide within the sequence, and the percent each nucleotide makes up of the total. You should include your output file in the submission in addition to our usual submission format (console output + code).

You should define 3 functions:

1. A function that reads the DNA sequence from a given filename and computes counts for each nucleotide. The function returns true if the operation was successful and false otherwise. **Hint:** the calculation of the counts happens in your function but the main function needs to know the results.
2. A function that outputs a formatted line for a nucleotide to a given output stream. This function is the only place in the code that knows how the line is formatted. **Hint:** you have to use `ostream&` type parameter to pass output streams, including `cout`.
3. A function that outputs the entire report to a given output stream. This function needs to call function 2 to output a line for each nucleotide.

The point of this assignment is to move your existing code into functions and figure out how to parameterize it, similarly to what I did in the last week's videos. You don't need to write too much new code.

**Your program should have the following:**

- Include 4 comment lines at the top: description of the program, author, section, and date. (1 point)

- Create your variables, use the appropriate type, name them appropriately, and remember to not leave them uninitialized. (1 point)
- Create named constants for what's appropriate. Decide the scope for each constant. Use the standard convention for constant names. (2 points)
- Define the 3 functions as described above.
- Function 1: (3 points)
  - Open the file for input and make sure the operation was successful
  - Read in nucleotides from the file and calculate counts for each one; if you encounter anything other than A, T, C, or G output an error message, skip over it, and keep going
  - Return true if the operation was successful and false otherwise
- Function 2: (2 points)
  - The function takes all needed parameters to output a single line of formatted output to an output stream. The output stream is also passed to it as a parameter. The type of this parameter needs to be `ostream&`.
  - How the output is formatted is the responsibility of this function. If you need to change it you shouldn't have to go to any other place in the code.
- Function 3: (3 points)
  - The function takes all needed parameters to output the entire report to an output stream. Similarly to function 2, the output stream is also passed to it as a parameter of type `ostream&`.
  - What goes into the report is the responsibility of this function. If you need to change anything about the report (other than the formatting of the lines – that's the responsibility of function 2) you shouldn't have to go to any other place in the code.

#### Notes:

- Pay attention to where you create and how you initialize your variables. Unsafe code is a 1-point deduction.
- Comment your code. Uncommented code is a 1-point deduction.
- Remember to include your output file in the submission. An insufficient submission is a 1-point deduction.
- Style: reference is a part of the type and needs to go with the type, e.g., `int& x` rather than `int &x`. Even though the language allows both

notations, the latter is more error-prone, needs to be avoided, and will carry a **0.25-point** deduction if used.

**Example Output:**

DNA sequence analysis:  
29782 nucleotides in the sequence

Sequence breakdown:

Adenine:	8892	29.86%
Thymine:	9581	32.17%
Cytosine:	5462	18.34%
Guanine:	5847	19.63%