

CS250 Assignment 3: DNA Set

Date assigned: Monday, February 25, 2019

Date due: Friday, March 8, 2019

Points: 40

Goals for this assignment

1. Read and use existing code.
2. Write an object-oriented program using two classes.
3. Use composition and friends.
4. Practice modular programming by using well-defined functions.

For this assignment, you will be using the DNA class that you created for your last assignment.

Step 1:

- Test the DNA project that you submitted for assignment 2 using the test cases provided on Grace. Fix any issues that you have so that all of the test cases run correctly.
- Add a function named **equals** to your DNA class that accepts a DNA object and returns true if the passed in DNA object is identical to the current DNA object, otherwise the function will return false.
- Add code to your driver to thoroughly test the **equals** function.

Step 2:

Create a new project, named 03_DNASet, in your assignment solution. This new project must be linked to the 02_DNAClasses project and be linked to the DNA.obj file produced by that project. All of the work below will happen in 03_DNASet.

Create a class, DNASet, which represents a set of DNA objects. This class will be similar to RationalSet.

You must provide the following functionality for the set:

- Provide an appropriate default constructor.
- Add a DNA strand to the set. The set can contain a maximum of 100 DNA objects. A set cannot contain the same strand more than once.
- Get the current number of items in the set.
- Determine if a particular DNA strand is in the set.
- Provide the set_union operation.
- Provide the set_intersection operation.
- Provide a way to retrieve a specific DNA strand from the set. The user will provide an integer, 0 to the number of items in the set -1, and you will provide the correct DNA strand.
- Provide a read function to read up to 100 DNA objects from the stream. Each DNA object is separated by whitespace.
- Provide a print function to print each DNA object to the stream, separated by a *space* (do not print a newline).
- You may add other operations as necessary.

Data

You must copy two data files, Pool1.txt and Pool2.txt, from Grace to the 03_DNASet project.

Driver

The driver for this project must:

1. Read each file (Pool1.txt and Pool2.txt) into its own DNASet.
2. Print each set to the screen.
3. Determine which DNA strands appear in both sets and print those strands to the screen.
4. Determine the longest strand in each set and print each of those strands to the screen. In the case of a tie, use the strands closest to the top of the file.
5. Determine the one strand in Pool1 and one strand in Pool2 that are most similar to each other and print both strands to the screen. Remember, P-Stat can only handle two DNA strands of the same length. In the case of a tie, use the strands closest to the top of the file.

Notes on design:

- You are to use object-oriented design using composition.
- For the DNASet class you will need a .h and a .cpp. You will also need a driver (named 03_DNASetDriver.cpp) containing the main function.
- You may not change the DNA class provided to you.

Error Handling:

- You should be able to handle an empty file. This is not an error; reading an empty file merely produces an empty set.
- The files do not need to contain the same number of strands or contain strands of the same length.
- If the files contain any invalid characters, then your program should quit.

Sample Files (Each file must end with a blank line)

Pool1.txt

Pool2.txt

CAT

AAA

T

TAC

AAT

Sample Output

```
*****  
* DNA POOL ANALYZER *  
*****
```

Pool 1

CAT AAA T

Pool 2

TAC AAT

In both Pools

In at least one Pool

CAT AAA T TAC AAT

Longest Strand in each Pool

Pool1: CAT

Pool2: TAC

Most Similar:

P-Stat: 0.33

Pool1: CAT

Pool2: AAT

To complete this assignment, you must submit the following:

1. An electronic copy of your program on Grace

- a. Add a project named 03_DNASet to your assignment solutions folder. It is vital that you name your solution and your project correctly! Make sure that 03_DNASet is correctly linked to 02_DNAClasses.
- b. Type your program (fully documented/commented) into the project. You need to follow the coding standards from the CS250 course page.
- c. Pay attention to the example output. Your program's output must look **exactly** like the sample output. The spacing and newlines in your output must match exactly.
- d. Make sure that your program builds without errors & warnings and runs correctly. If you get any errors or warnings, double check that you typed everything correctly. Be aware that C++ is case-sensitive. You will lose 10% if there are any warnings and 40% if your program does not build successfully.
- e. Once you are sure that the program works, it is time to submit your program. You do this by logging on to Grace and placing your complete solution folder in the **CS250-0X Drop** folder.
- f. The solution must be in the drop folder by the time class starts on the day the assignment is due.

2. A hard copy of your program

- a. The hard copy must be placed on the instructor's desk by the time class starts on the day that it is due.
- b. **The hard copy must be printed in color, double-sided, and stapled in the upper left corner if your solution contains multiple pages.** Failure to print properly will result in loss of 4 points (10%)
- c. Your tab size must be set to 2 and you must not go past column 80 in your output.

Remember, if you have any problems, come to me straight away

with your project on Grace. Good Luck!!!!