

Program Requirements

In this assignment you will create a double-strand DNA pattern checking and correcting application. Information about the basics of DNA can be found at:

- <http://ghr.nlm.nih.gov/handbook/basics/dna>

In particular, the step-by-step procedure (i.e. algorithm) for this assignment is:

1. Prompt the user to enter a upper DNA strand
2. Read-in the string value entered by the user
3. Prompt the user to enter a lower DNA strand
4. Read-in the string value entered by the user
5. If the entered lower stand or upper strand is empty (i.e. contain no chemical bases) GOTO step 13, else GOTO step 6.
6. If the entered lower and upper strands have the same number of chemical bases GOTO step 7, else GOTO step 12
7. If the entered lower and upper strands only contain combinations of adenine (A), guanine (G), cytosine (C), and thymine (T) chemical bases then GOTO step 8, else GOTO step 11
8. If the upper and lower strands only define valid base-pairs (i.e. A paired with T, and C paired with G) then GOTO step 9, else GOTO step 10
9. Display to the user:
 - “The entered double-strand DNA pattern is correct ... Exiting program”.
10. Fix the base-pair errors (**your choice to fix error in upper or lower strand**) and display to the user:
 - “The entered double-strand DNA pattern had base-pair errors that have been corrected: ”,
 - The fixed upper strand in upper case with chemical base correction(s) in lower case,
 - The fixed lower strand in upper case with chemical base correction(s) in lower case, and
 - “... Exiting program”
11. Display to the user “The entered upper and lower stands must only contain combinations of A, G, C, or T ... Exiting program”.
12. Display to the user “The entered upper and lower strands do not have the same number of chemical bases ... Exiting program”.
13. Display to the user “The entered upper strand or lower strand is not defined ... Exiting program”.

Four different sample input and output scenarios are provided that demonstrate the above algorithm.

```
Enter upper DNA strand: AATCGGCA
Enter lower DNA strand: TTAGCCG
```

```
The entered upper and lower strands do not have the same number of
chemical bases ... Exiting program
```

```
Enter upper DNA strand: AATCGGCB
Enter lower DNA strand: TTAGCCGT
```

The entered upper and lower strands must only contain combinations of A, G, C, or T ... Exiting program

```
Enter upper DNA strand: AATCAGCA
Enter lower DNA strand: TCAGCCGT
```

The entered double-strand DNA pattern had base-pair errors that have been corrected: AATCgGCA TtAGCCGT

... Exiting program

```
Enter upper DNA strand: AATCGGCA
Enter lower DNA strand: TTAGCCGT
```

The entered double-strand DNA pattern is correct ... Exiting program

Program Design and Specification

- Only one class named **DNA** and contains a main method (no other methods are needed). Your entire solution must be implemented in the main method.
- The DNA class must be in a package named **edu.cofc.csci230**.
- To receive full credit, the completed program must fully implement the provided algorithm.
- Exception handling should not be included in this assignment.
- Hint: Use the methods (e.g., `charAt()`, `length()`, `isEmpty()`) in the String class.

Program Submission

Create a ZIP file that only contains the DNA.java file. For the ZIP file you **must** use the naming convention:

`<lastname>.zip`
e.g., Munsell.zip

If the assignment is not submitted in the correct format – it will not be accepted – no exceptions! Submit the ZIP file via OAKS in the Dropbox that corresponds to the assignment. Resubmit as many times as you like, the newest submission will be the graded submission.

Grading Rubric

20 Points	Style: Comments and Indentation
80 Points	Functionality: <ul style="list-style-type: none">• Program compiles (10)• Program runs (10)• For given inputs, program produces correct output (60)

If the submitted program does not compile: 0 of 80 points

If the submitted program compiles but does not run: 10 of 80 points

If the submitted program compiles and runs: 20 of 80 points

If the submitted program compiles, runs, and produces correct output: 80 of 80 points

The correctness of your program will be evaluated using test cases developed by the instructor.

Late assignments will not be accepted – no exceptions (please do not email me your assignment after the due date, I will not accept it).

Please feel free to setup an appointment to discuss the assigned problem. I'll be more than happy to listen to your approach and make suggestions. However, I cannot tell you how to code the solution. Furthermore, code debugging is your job.