

Lab Assignment Week 09

CSC/DSCI 1301 – Principles of CS/DS I

Week of March 10th, 2025

Introduction

Welcome to the ninth programming lab of CSC/DSCI 1301! Today we will be covering the following topics:

- Using Default Argument Values
- Using Arbitrary Arguments
- String Methods

Lab policy reminders:

- Attendance is mandatory.
- Labs must be completed **individually**.
- TAs are here to help you. Ask them for help!
- Lab assignments are due at the end of each lab.

Comments

The lab assignment requires the inclusion of comments to enhance code readability and understanding. Specifically, a block comment at the beginning of the Python file is required. Your block comment should include the following:

- The program name
- The author's name (your name)
- A description of the program's overall purpose

Additionally, inline comments should be used throughout the code to explain specific lines or sections that might be less obvious to someone reading the code. These inline comments can clarify complex calculations, explain the purpose of certain variables, or provide additional context for specific code blocks.

Deliverables:

1. Python files for your DNA translation program.
 - a. dna_translation.py
 - b. helper.py
2. Screenshots of your program output

If you have any questions, please do not hesitate to ask your TA!

Program 1: dna_translation.py

Computers are extremely efficient at storing and manipulating large data. For this Lab, you will need to write a program that transcribes a DNA sequence into mRNA, then translates that mRNA sequence into the sequence of amino acids that make up the protein. You will need to write two functions, **transcription()** and **translation()**, to accomplish this task. Your program should also print out the sequence after each step.

DNA to mRNA

The **transcription()** function takes a DNA sequence as input, transcribes it into the complementary mRNA sequence and returns the mRNA sequence as output. You will need to program your **transcription()** function to perform two tasks:

1. Replace the Nucleotide Thymine (T) with Uracil (U).
2. Create the Base Pair Complement of the sequence.

The rules for creating the Base Pair Complement are shown in the table below.

RNA Base Paring Rules Table

Nucleotides	Codons
Adenine to Uracil	A-U
Uracil to Adenine	U-A
Cytosine to Guanine	C-G
Guanine to Cytosine	G-C

Hints

- The built-in String method **replace()** will replace the Thymine (T) with Uracil (U) DNA string.
 - The **replace()** method returns the string after replacement.
 - You can read more about this method [here](#).
- However, the String **replace()** method won't be suitable for creating the Base Pair Complement String.
 - Use a similar method to the `text_filter.py` program to swap the RNA Base Pairs using a for loop and if-statements.

mRNA to Protein

Protein translation comprises two steps. The mRNA is broken up into nucleotide triplets. The nucleotides that comprise the triplet determine which amino acid makes up that part of the protein.

The **translation()** function will take in the mRNA sequence as input, translate it into a sequence of amino acids, and return the amino acid sequence as its output. Your **translation()** function will need to do the following:

1. Separate the mRNA sequence into nucleotide triplets.
 - a. **Note: There is no built-in method that does this. You will have to write a helper function that separates a string into triplets: a list of strings containing 3 letters.**
2. Look up the amino acid that corresponds to the nucleotide triplet.
 - a. The provided helper .py module contains a dictionary of amino acids.
 - i. The keys for the dictionary are the nucleotide triplets shown in the chart below.
 - ii. The values are the 3-letter abbreviations of the amino acids.

Nucleotide to Amino Acid Chart

		Second nucleotide					
		U	C	A	G		
First nucleotide	U	UUU Phe	UCU	UAU Tyr	UGU Cys	Third nucleotide	U
		UUC	UCC Ser	UAC	UGC		C
		UUA Leu	UCA	UAA STOP	UGA STOP		A
		UUG	UCG	UAG STOP	UGG Trp		G
	C	CUU	CCU	CAU His	CGU		U
		CUC Leu	CCC Pro	CAC	CGC Arg		C
		CUA	CCA	CAA Gln	CGA		A
		CUG	CCG	CAG	CGG		G
	A	AUU	ACU	AAU Asn	AGU Ser		U
		AUC Ile	ACC Thr	AAC	AGC		C
		AUA	ACA	AAA Lys	AGA Arg		A
		AUG Met	ACG	AAG	AGG		G
	G	GUU	GCU	GAU Asp	GGU		U
		GUC Val	GCC Ala	GAC	GGC		C
		GUA	GCA	GAA Glu	GGA		A
		GUG	GCG	GAG	GGG		G

Hint

- Step 1:
 - Create an Empty List
 - Iterate over the mRNA string
 - Keep track of the last 3 characters
 - Every multiple of 3 characters, append the last 3 characters as a string into the list.
- Step 2:
 - Use the same method as last week's `text_filter.py` program to look up the triplets in the dictionary to build the final protein sequence.

Example Output

DNA Sequence

TACGCAGAAAAAATCAGCGGGGTTGTTGGTCATTAGTCTGAATT

mRNA Sequence

AUGCGUCUUUUUUAGUCGCCCCAACCAACAGUAAUCAGACUUAA

Protein Sequence

Met Arg Leu Phe Leu Val Ala Pro Thr Thr Ser Asn Gln Thr STOP

Skills Covered

- Creating User-defined Functions
- String Methods

Deliverables

For this program you will need to provide the python file containing your code as well as a screenshot of the output of your program. Please name your files as follows:

- Python Files
 - lastname_firstname_filename.py
 - For example: **hawamdeh_faris_dna_translation.py**
- Screenshots
 - lastname_firstname_filename.png
 - For example: **hawamdeh_faris_dna_translation.png**