# CSC121 Lab 07: More About Strings

## Goals

In this lab assignment, students will demonstrate the ability to:

* Create and use strings
* Use slicing to select ranges of characters in strings
* Use string methods and operators

NOTE: This lab document is a copyrighted work of Wake Tech Community College and the course instructor. Any posting of this document outside of Wake Tech is considered a copyright violation. Students who post these documents outside of Wake Tech are subject to academic and possible legal actions.

## Instructions

In this lab, you will demonstrate your mastery of the string data type.

Follow the instructions in each problem and submit the specified files.

Problem 1 will consist of programs that you create from scratch that meets the problem specification. In Problem 2, you'll be provided starter code that you must complete.

## Problems

### Problem 1

This program will test your ability to examine and manipulate strings.

Create a file named **Lab07P1.py**. Write a program that does the following:

Use a loop to ask the user to enter a series of IP addresses in a specific format: ###.###.###.###. For example, 192.168.0.1.

You will need to validate that the IP address is in the correct format. Check for these validation conditions IN THIS ORDER:

* Split what the user enters across the periods and see if there are 4 separate parts. If there are not, display an appropriate error message and continue in the loop. Do not check on other error conditions.
* Check whether each part of the IP address is a number. If any part is not a number, display an appropriate error message and continue in the loop without checking other error conditions.
* Check whether each number in the IP address is between 0 and 255 inclusive. If any number is not, display an appropriate error message and continue in the loop without checking other error conditions.

If the IP address entered was valid, then change the periods to colons and display the newly formatted IP address.

For example, if the IP address was: 192.168.0.1

…the program should display: 192:168:0:1

The program will continue asking the user for IP addresses until they enter a q or Q to exit the program.

Here's pseudocode to help you solve this problem:

# Ask the user to input an IP address or 'Q' to quit

# while the user has not entered 'Q' or 'q' to quit

# split the user input at each period and store the parts in a list

# if there are not 4 parts in the list

# display an error message

# else

# set the variable error\_flag to False

# for each part in the list

# if the part is not a number or if it is not between 0 and 255

# display an error message

# set error\_flag to True

# break the loop

# if no error has been displayed (i.e., error\_flag is False)

# replace each period in the user input with a colon

# display the new formatted IP address

# ask the user to input an IP address or 'Q' to quit

There are many different ways that this program could be implemented, HOWEVER:

**You are required to implement the program as directed by the pseudocode.**

Any other implementation will be penalized with a 25% reduction (15 points) in grade for this problem.

Sample output:

Please enter an IP address or 'Q' to quit: 122.44.33

Error: An IP address should consist of 4 parts separated by periods.

Please enter an IP address or 'Q' to quit: 122.44.33.x

Error with x: Each part of the IP address should be a number between 0 and 255.

Please enter an IP address or 'Q' to quit: 122.444.33.x

Error with 444: Each part of the IP address should be a number between 0 and 255.

Please enter an IP address or 'Q' to quit: 122.44.33.89

Reformatted IP address: 122:44:33:89

Please enter an IP address or 'Q' to quit: q

Submit the program file **Lab07P1.py** to Blackboard for credit. No SCREENSHOTS should be submitted for this lab or future labs unless specifically requested.

### Problem 2

This program also tests your ability to analyze and manipulate strings.

You have been provided a starter file called **Lab07P2-starter.**py. Download that file from Blackboard and rename it **Lab07P2.py**.

The starter code will open a file named strings.txt if it is in the same directory as your Python program. You can create your own stings.txt file or download the one provided on Blackboard.

*NOTE: Opening and reading files is something you will learn about in Lesson 8.*

The starter code will read the file line by line, and pass each line to a function called manipulate\_text(). You will be filling in the code for this function.

The manipulate\_text() function accepts one string as input. The function should do the following with the string parameter:

* Strip the leading and trailing whitespace, and output the string.
* Replace all occurrences of $NAME with your first name.
* Replace all occurrences of $EMAIL with your email address.
* Replace all occurrences of $CITY with the name of the city where you live.
* Print the updated line.
* Print a message indicating the number of characters in the updated line.
* Count the number of occurrences of your first name and print a message reporting the count.
* Use floor division to divide the number of characters by 2, then print the first half and last half of the line.
* Print the line in uppercase.
* Print the line in lowercase.

For example, if this is the contents of strings.txt:

$NAME, $EMAIL, $CITY

My name is $NAME

My email is $EMAIL

… and your first name was “StudentName”, your email was student@waketech.edu, and your city was “StudentCity, this would be the output of your program:

Original line: $NAME, $EMAIL, $CITY

Updated line: StudentName, student@waketech.edu, StudentCity

Number of characters in updated line: 46

Occurrences of 'StudentName': 1

First half of line: StudentName, student@wa

Second half of line: ketech.edu, StudentCity

Line in uppercase: STUDENTNAME, STUDENT@WAKETECH.EDU, STUDENTCITY

Line in lowercase: studentname, student@waketech.edu, studentcity

Original line: My name is $NAME

Updated line: My name is StudentName

Number of characters in updated line: 22

Occurrences of 'StudentName': 1

First half of line: My name is

Second half of line: StudentName

Line in uppercase: MY NAME IS STUDENTNAME

Line in lowercase: my name is studentname

Original line: My email is $EMAIL

Updated line: My email is student@waketech.edu

Number of characters in updated line: 32

Occurrences of 'StudentName': 0

First half of line: My email is stud

Second half of line: ent@waketech.edu

Line in uppercase: MY EMAIL IS STUDENT@WAKETECH.EDU

Line in lowercase: my email is student@waketech.edu

Submit the program file **Lab07P2.py** to Blackboard for credit.

## Grading Rubric

### Grading rubric for Problem 1 (60 points)

* Program has a well-formatted and correct header [5 points]
* Program does execute correctly and produces correct results [55 points]
* 15 point deduction if program is not implemented with given psudocode.

### Grading rubric for Problem 2 (40 points)

* Program has a well-formatted and correct header [5 points]
* Program does execute correctly and produces correct results [35 points]