\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC121 PYTHON Programming**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 10 Text Files and Strings

# Objectives

In this lab assignment, students will learn:

- How to write code to read from a text file

- How to write code to write to a text file

- How to apply sequence operations on strings

- How to use string methods

# Goals

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

- Write code to read from a text file

- Write code to write to a text file

- Apply sequence operations on strings

- Use string methods

# Instruction and Problems

Write a Python program for each of the problems in this lab. Please use PyCharm to type and test your programs.

## Problem 1

Write a program to store water usage data of 4 customers in a text file. The program asks the user to enter account number, customer type (R for residential or B for business), and number of gallons used for each of the 4 customers. Store the data in a text file named “water.txt”. Overwrite old data in “water.txt” if the file already exists. The following is an example.

Enter account number: 3147

Enter R for residential, B for business: R

Enter number of gallons used: 1000

Enter account number: 2168

Enter R for residential, B for business: R

Enter number of gallons used: 6002

Enter account number: 5984

Enter R for residential, B for business: B

Enter number of gallons used: 2000

Enter account number: 7086

Enter R for residential, B for business: B

Enter number of gallons used: 8002

The example above writes the following data to the file “water.txt”:

3147 R 1000

2168 R 6002

5984 B 2000

7086 B 8002

Save your Python program in a file named **Lab10P1.py**. Submit the file for credit.

## Problem 2

Write a program to read data stored in “water.txt”, which was created in Problem 1. Calculate water charge for each customer. Residential customers pay $0.005 per gallon for the first 6000 gallons. If the usage is more than 6000 gallons, the rate will be $0.007 per gallon after the first 6000 gallons. Business customers pay $0.006 per gallon for the first 8000 gallons. If the usage is more than 8000 gallons, the rate will be $0.008 per gallon after the first 8000 gallons. Display account number and water charge of each customer on computer screen.

The following is an example.

Account number: 3147 Water charge: 5.0

Account number: 2168 Water charge: 30.014

Account number: 5984 Water charge: 12.0

Account number: 7086 Water charge: 48.016

Save your Python program in a file named **Lab10P2.py**. Submit the file for credit.

## Problem 3

Write a program to do the following. Ask user to enter time in the format of hh:mm:ss. For example, 11:07:28. Hour must be a two-digit number between 0 and 23, inclusive. Minute and second must be two-digit numbers between 0 and 59, inclusive. The program needs to check whether the time entered is valid.

1. Check whether there are exactly two colons. Display an error message and stop if the input is invalid.
2. Check whether hour is a two-digit number. Display an error message and stop if the input is invalid.
3. Check whether minute is a two-digit number. Display an error message and stop if the input is invalid.
4. Check whether second is a two-digit number. Display an error message and stop if the input is invalid.
5. Check whether hour is between 0 and 23, inclusive. Display an error message and stop if the input is invalid.
6. Check whether minute is between 0 and 59, inclusive. Display an error message and stop if the input is invalid.
7. Check whether second is between 0 and 59, inclusive. Display an error message and stop if the input is invalid.
8. If the time entered is valid, remove the colons and display the time. For example, if the input time is 11:07:28, the program should display 110728.

The following are a few examples:

Enter time [hh:mm:ss]: 10.44.58

Must separate hour, minute and second with colons.

Enter time [hh:mm:ss]: 11:27 28

Must separate hour, minute and second with colons.

Enter time [hh:mm:ss]: 11:16:ss

Second must be a 2-digit number.

Enter time [hh:mm:ss]: 15:4:26

Minute must be a 2-digit number.

Enter time [hh:mm:ss]: 24:05:07

Hour must be a 2-digit number between 0 and 23.

Enter time [hh:mm:ss]: 14:07:28

Time with colons removed: 140728

Save your Python program in a file named **Lab10P3.py**. Submit the file to Blackboard for credit.

## Problem 4

Write a program to do the following. Ask the user to enter a string. Convert all letters to uppercase. Count and display the frequency of each letter in the string. For example, suppose the user enters the string “Csc.565”, the program should display the following output because the letter C appears twice while the letter S appears once:

C 2

S 1

Each letter cannot be shown more than once in the output. The following output is incorrect because the letter C is shown twice:

C 2

S 1

C 2

The program should count letters only but no other characters. The following output is incorrect because it counts non-letter characters:

C 2

S 1

C 2

. 1

5 2

6 1

The following is an example.

Enter a string: Magee, Mississippi

M 2

A 1

G 1

E 2

I 4

S 4

P 2

Save your Python program in a file named **Lab10P4.py**. Submit the file to Blackboard for credit.

# Grading rubric for Program 1

Open and close text file [10 points]

Write data to text file [10 points]

Other statements [5 points]

# Grading rubric for Program 2

Open and close text file [10 points]

Read data from text file [10 points]

Other statements [5 points]

# Grading rubric for Program 3

Check whether time entered is valid [20 points]

Remove colons from string [10 points]

# Grading rubric for Program 4

Count frequency of each letter in the string [20 points]