CSCI 1411: Fundamentals of Computing
Lab 10
Due Date: November 3, 2023

Name: _____

Goals:

- To understand how to write user-defined functions.
- To understand details of function calls and parameter passing in Python.

Development Environment: IDLE

Deliverables:

1. This completed document with required screen shots.
2. Python file created for the lab. Name the file using the following format: `lastnameLab10a.py` and `lastnameLab10b.py`.

How to take a **screen shot**:

- For a Windows 10: Use Snipping Tool to copy and press CTRL + V to paste screen shot.
- For Mac: Press Shift + Command (⌘) + 4 to copy and press Command (⌘) + V to paste screen shot.

Part I – Skill Practice

Start IDLE and create a new file. Type the following code in the file. …..**Do not cut and paste.** You will learn more by typing it in. In this lab, we have provided three functions as follows:

- First function converts the temperature in degree Fahrenheit to temperature in degree Celsius and returns the temperature in degree Celsius. It shows how to pass data to a function using a parameter and how to return a value using the return statement.
- The second function is the example which shows how to pass a List object as a parameter. This function also demonstrates that if you pass a list object as a parameter then it can be changed by the function as list is a mutable object.
- The third function is the main function. This function invokes the first two functions.

```
# Name:
# Class: CSCI 1411-00x
# Due Date
# Description: This program how to use functions in a Python
# program. It also shows how to use parameters to pass data
# to a function and return data from a function using a return
# statement.

# f_to_c function
# parameter: Temperature in degree Fahrenheit (fahrenheit)
# returns equivalent temperature in degree Celsius
def f_to_c(fahrenheit):
    result = (fahrenheit - 32) * (5.0 / 9.0)
    return result

# f_to_c_list function
# parameter: a list of temperature in degree Fahrenheit(temp_list)
# It coverts the temperatures in list to equivalent temperatures in
# Celsius. There is no return statement as data is returned using
# the list that is passed as an argument.
def f_to_c_list(temp_list):
    for i in range(len(temp_list)):
        fahrenheit = temp_list[i]
        celsius = f_to_c (fahrenheit)
        temp_list[i] = round(celsius,2)

# main function
def main():
    temp_list = []
    # Ask user for a set of 5 temperature in degree Fahrenheit
    for i in range(5):
        fahrenheit = float(input("Enter temperature in degree Fahrenheit: "))
        temp_list.append(fahrenheit)

    # call the function f_to_c_list
    f_to_c_list(temp_list)
    print("The converted temperature list")
    print(temp_list)
```

Save the file as `lastnameLab10a.py`

Run your program and test it using the data in the following table.

Run Num	Input					Output
	Temp 1	Temp 2	Temp 3	Temp 4	Temp 5	
1	0	10	20	30	40	[-17.78, -12.22, -6.67, -1.11, 4.44]
2	-40	32	45	55	212	[-40.0, 0.0, 7.22, 12.78, 100.0]

Sample I/O

```
>>> main ()
    Enter temperature in degree Fahrenheit: -40
    Enter temperature in degree Fahrenheit: 32
    Enter temperature in degree Fahrenheit: 45
    Enter temperature in degree Fahrenheit: 55
    Enter temperature in degree Fahrenheit: 212
    The converted temperature list
    [-40.0, 0.0, 7.22, 12.78, 100.0]
```

Take a screen shot of your results and paste it in the box below:

Screen Shot 1

Part II – Convert the Date format

Problem Statement: Write a program that will convert the date from short format ('mm/dd/yyyy') to long format ('Month day, year'). For example, date '01/23/2023' will be converted to January 23, 2023. Start by writing the code for the following functions (feel free to test the functions as you write them):

`month_to_name` function: This function will convert the month number to the month name. The month number will be passed as a parameter and it will return the corresponding month name. For example, if the value of parameter is '01' then it will return January. Note that the month number is passed as a string. Hint: You can store month names in a list and use the parameter value (converted to int) as index to find the corresponding month name.

`split_date` function: This function will take the short date with the format 'mm/dd/yyyy' and will return the mm, dd, and yyyy. Short date will be passed as a parameter and it will use the return statement to return dd, mm, yyyy. For example, if the value of the parameter is '01/12/2023' then it will return '01', '12', and '2023'. You can use the split method in the string class to split the short date into the required three parts as follows:
month, day, year = short_date.split ('/') where short_date is the name of the parameter for the function. Note that it returns three strings using one return statement.

`date_convert` function: This function will convert the date given in the short format to the date in the long format. It will receive the date in short format, which is a string, as a parameter. It will use the split_date function to split the date given in short format (mm/dd/yyyy) into its components (mm, dd, and yyyy). It will use the month_to_name function to convert the mm into its name. It will combine the name of the month, dd and yyyy to create a new string with date in long format. It will return this date using the return statement.

`main` function: This function will ask the user to enter the date using short format (mm/dd/yy), it will use the date_convert function to convert the date into long format and will display the date in the long format.

Sample I/O:

```
>>> main ()
    Enter date: 01/22/2023
    01/22/2023 in long fomat is January 22, 2023
>>> main ()
    Enter date: 10/23/2024
    10/23/2024 in long fomat is October 23, 2024
>>>
```

Run your program and take a screen shot of your result and paste it in the box below:

Screen Shot 2

Every program should have the following comment block at the top. Make sure to fill in your name, class with section number, due date, brief description of your program, and status of your program:

```
#
# Name:
# Class: CSCI 1411-00X
# Due Date:
# Description:
# Status:
```

Rubric for Lab 10:

Criteria	Rating
Part I (Screen shot 1)	Screen shot included – 5 points No screen shot included – 0 points
Part I: Python Program	Python program is submitted – 5 points Python program is not submitted – 0 points
Part II (Screen shot 2)	Screen shot included – 5 points No screen shot included – 0 points
Part II: Python Program – month_to_name function	Function works as expected – 20 points Function does not work as expected – 0 points
Part II: Python Program – split_date function	Function works as expected – 20 points Function does not work as expected – 0 points
Part II: Python Program – date_convert function	Function works as expected – 20 points Function does not work as expected – 0 points
Part II: Python Program – date_convert function	Uses the split_date and month_to_name functions to perform the sub-tasks – 5 points Does not use split_date and month_to_name functions – 0 points
Part II: Python Program – main function	Prompt for and reads in the date in short format – 5 points Does not read in the date in short format – 0 points
Part II: Python Program – main function	Uses date_convert function to convert date from short format to long format – 5 points Does not use date_convert function to convert date from short format to long format – 0 points
Part II: Python Program – main function	Displays the date in long format using appropriate label (see sample I/O) – 5 points Displays the date in long format without any labels – 0 points
Total Points	95