**CSE1100 – Programming Concepts with Python**

**Programming Concepts Lab Report**

**Fall 2021**

*By*

Cael Shoop

*Computer Science, B.S.*

*cshoop2018@my.fit.edu*

November 6, 2021

Lab #4

Teaching Instructor:

Stefan Joe-Yen, Ph.D.

# 1. Problem Statement

Problem 1

This problem was the need for the improvement of our first Lab 2 problem via the implementation of input checks, as well as adding profile creation and saving through a main menu.

Problem 2

This problem was up to us to decide. I chose, after looking at all of the more recent programs I had written for this class, to modify Lab 2’s second problem. This required the input of 3 numbers with spaces between each, and I had previously implemented none of the checks that it needed.

# 2. Requirements

Problem 1

1. The program must have a main menu with three selections.
   1. Enter & Save Data
   2. Load & Show Data
   3. Quit
2. Enter & Save Data
   1. This option will contain the base program.
   2. This option will also allow the user to choose whether or not they want to save the profile.
   3. The profile shall be saved under a folder named “profiles,” with the filename being the profile’s first and last name followed by “.txt.”
3. Load & Show Data
   1. This option will load all saved profiles and display them on the screen.
   2. It shall display the data in the same way the base program did.
4. Quit
   1. This option shall terminate the program with no errors.

Problem 2

1. The program shall take 3 numbers as input.
   1. There must be input checks in place to ensure the input consists of 3 numbers.
2. The program will convert these inputs to floats and calculate and output the highest, lowest, and average values.

# 3. Software Construction (Annotated Python Code)

Problem 1

'''Written by Cael Shoop. CSE1100 Lab 4 Problem 1, originally Lab 2 Problem 1.'''

from math import floor

import glob

import os

class profile():

# Create user profile

def \_\_init\_\_(self, firstName, lastName, height):

self.firstName = firstName

self.lastName = lastName

self.height = height

# Display user profile

def show(self):

print('\tFIRST NAME: ' + self.firstName)

print('\tLAST NAME: ' + self.lastName)

print('\tHEIGHT: ' + self.height)

# Save user profile

def save(self):

success = False

while not success:

try: # If there is a syntax error, the program will not crash.

action = input('Would you like to save this profile? (Y/N)\n')

if action == 'Y' or action == 'y':

print('Saving profile...')

filename = 'profiles/' + self.firstName + self.lastName + '.txt'

file = open(filename, 'w')

try: # Writes to the file

file.write(self.firstName + '\n' + self.lastName + '\n' + self.height)

file.close()

print('Profile saved!\n')

success = True

except:

print('Failed to write to file.\n')

elif action == 'N' or action == 'n':

print('Profile will not be saved.\n')

success = True

else:

print('Please enter \'Y\' or \'N\'.')

except:

print('Please enter \'Y\' or \'N\'.')

# Show all profiles

def show\_profiles():

print('\nHere are the current stored profiles:\n')

path = 'profiles/' # Reads all .txt files within profiles folder

for filename in glob.glob(os.path.join(path, '\*.txt')):

with open(os.path.join(os.getcwd(), filename), 'r') as f:

firstName = f.readline().strip('\n') # Reads each line as each variable

lastName = f.readline().strip('\n')

height = f.readline().strip('\n')

current = profile(firstName, lastName, height)

current.show()

print('')

# Convert raw height in inches to feet and inches

def convert(height):

heightFeet = floor(height / 12)

heightInches = height % 12

heightstr = str(heightFeet) + '\' ' + str(heightInches) + '"\n'

return heightstr # Returns the formatted height string

# Reads in user info with checks

def takeInfo():

success = False

while (not success):

try: # First and last name input checking

firstName, lastName = input('Hello! What is your full name?\n').split(' ')

if not firstName or not lastName:

print('Please enter your first and last name with a space in between.')

else:

success = True

except:

print('Please enter your first and last name with a space in between.')

success = False

while (not success):

try: # Height input checking

height = int(input('What is your height (in inches)?\n'))

if not height:

print('Please enter your height in numerical form, in inches.')

else:

success = True

except:

print('Please enter your height in numerical form, in inches.')

return (firstName, lastName, height)

# Displays the main menu and returns a checked input selection

def menu():

print('(Main Menu) Please select an action:')

print('\t(1) Enter & Save Data')

print('\t(2) Load & Show Data')

print('\t(3) Quit')

success = False

while (not success):

try:

action = int(input())

if not action or action < 1 or action > 3:

print('Please enter a valid menu selection number.')

else:

success = True

except:

print('Please enter a valid menu selection number.')

return action

def main():

repeat = True

while (repeat):

choice = menu()

if choice == 1: # Input & optionally save data

firstName, lastName, heightraw = takeInfo()

height = convert(heightraw)

current = profile(firstName, lastName, height)

print('Great to meet you, ' + current.firstName + '! Here is your profile:\n')

current.show()

current.save()

elif choice == 2: # Display all saved profiles

show\_profiles()

else: # Quit

repeat = False

if \_\_name\_\_ == '\_\_main\_\_':

try:

main()

except:

print('Error. Main failed to execute correctly.')

else:

print('Please run this script directly.')

Problem 2

'''Written by Cael Shoop. CSE1100 Lab 4 Problem 2, originally Lab 2 Problem 2.'''

from math import \*

# Runs input checks to ensure all input is valid,

# then returns a list of the inputs as float values.

def secure\_input():

success = False

while not success:

try: # Input checking

num0, num1, num2 = input('Please enter three different numbers\n').split(' ')

if not num0 or not num1 or not num2:

print('Please enter three numbers separated by spaces. e.g. \'3 8 7\'')

else:

try:

nums = []

nums.append(float(num0))

nums.append(float(num1))

nums.append(float(num2))

success = True

return nums

except:

print('Please enter three numbers separated by spaces. e.g. \'3 8 7\'')

except:

print('Please enter three numbers separated by spaces. e.g. \'3 8 7\'')

def main():

# Input and conversion to list

nums = secure\_input()

# Calculate average

avg = sum(nums) / len(nums)

# Output

print('\nThanks. Here are the results:')

print('\tMIN: ' + str(min(nums)))

print('\tMAX: ' + str(max(nums)))

print('\tAVG: ' + str(avg) + '\n')

if \_\_name\_\_ == '\_\_main\_\_':

try:

main()

except:

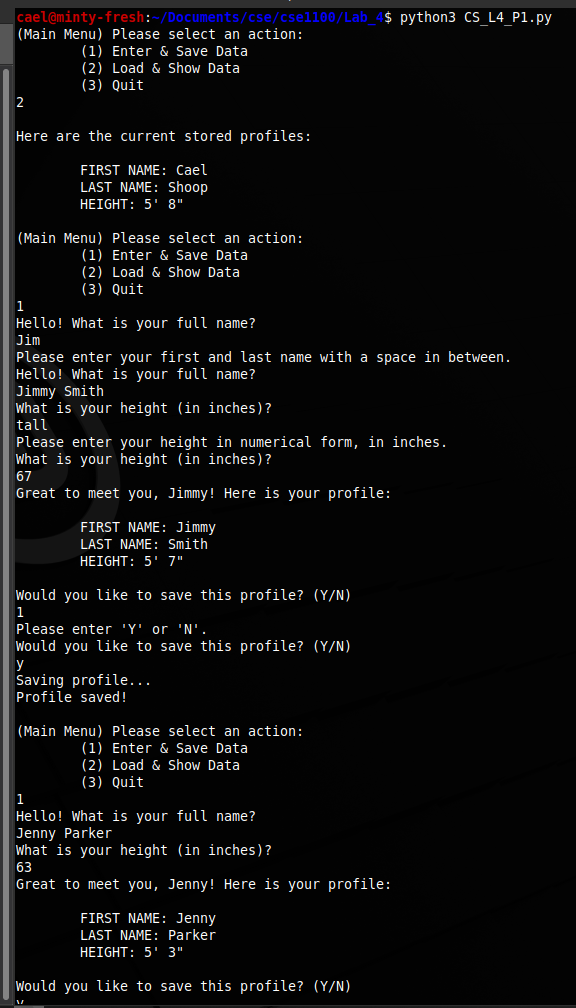
print('Error. Main failed to execute correctly.')

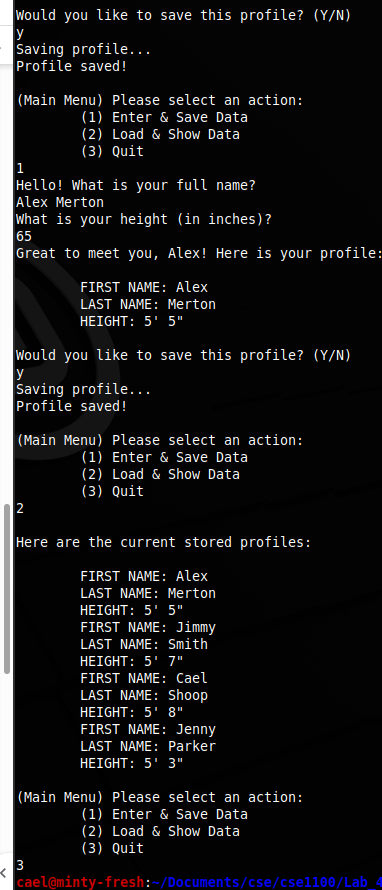
else:

print('Please run this script directly.')

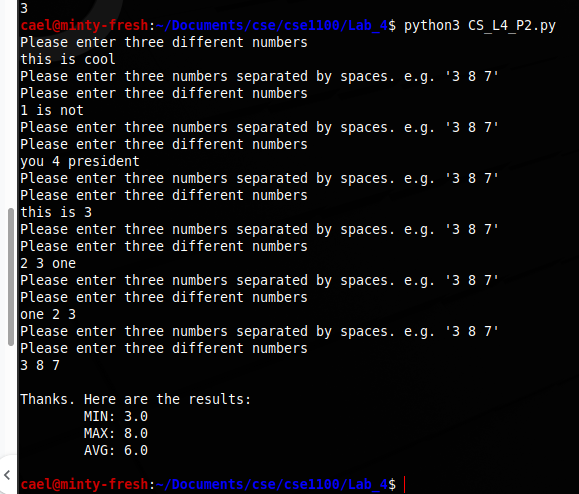
# 4. Software Testing

Problem 1





Problem 2



# 5. Self-Reflection

I got to learn about using class structures and self, along with returning multiple variables at once. These will be very useful as I have been using class structures in my python code for senior project, specifically for Roomba information. I will continue to use it for the Roomba simulation we may end up making.