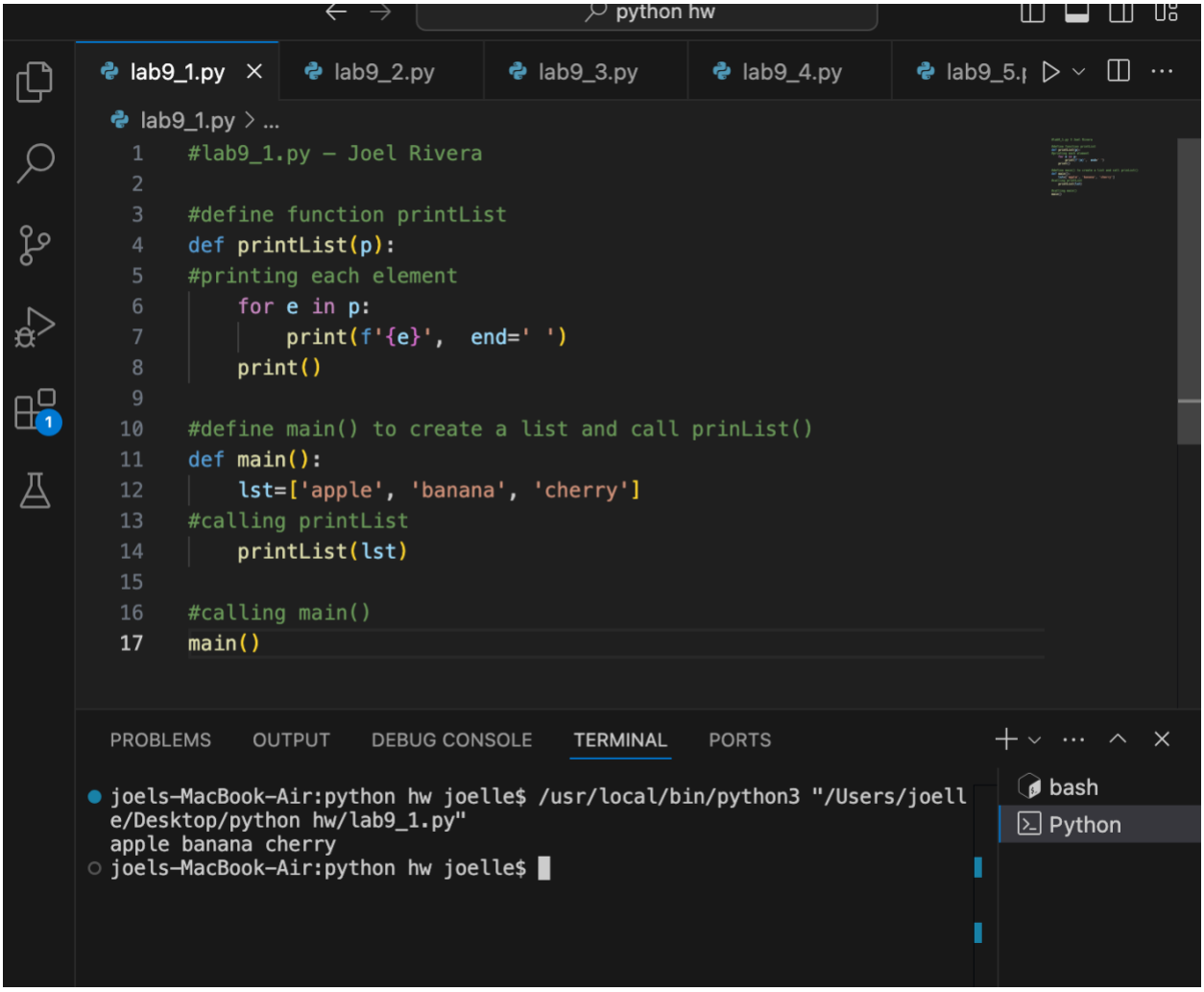
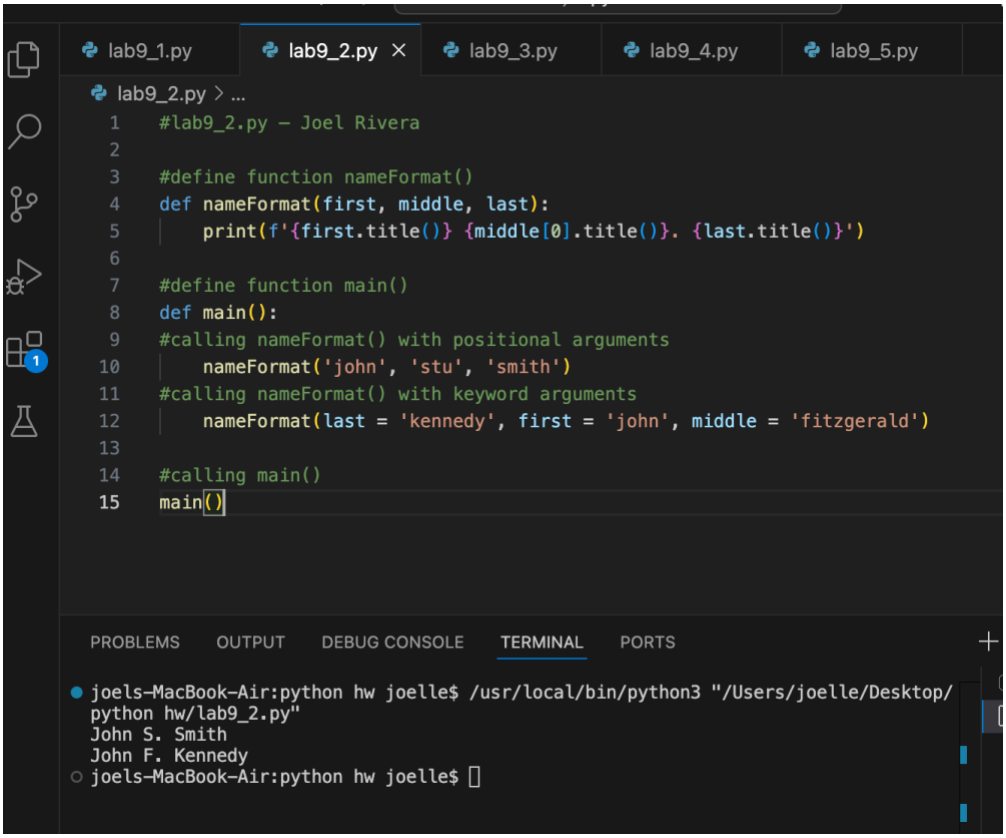
Joel Rivera

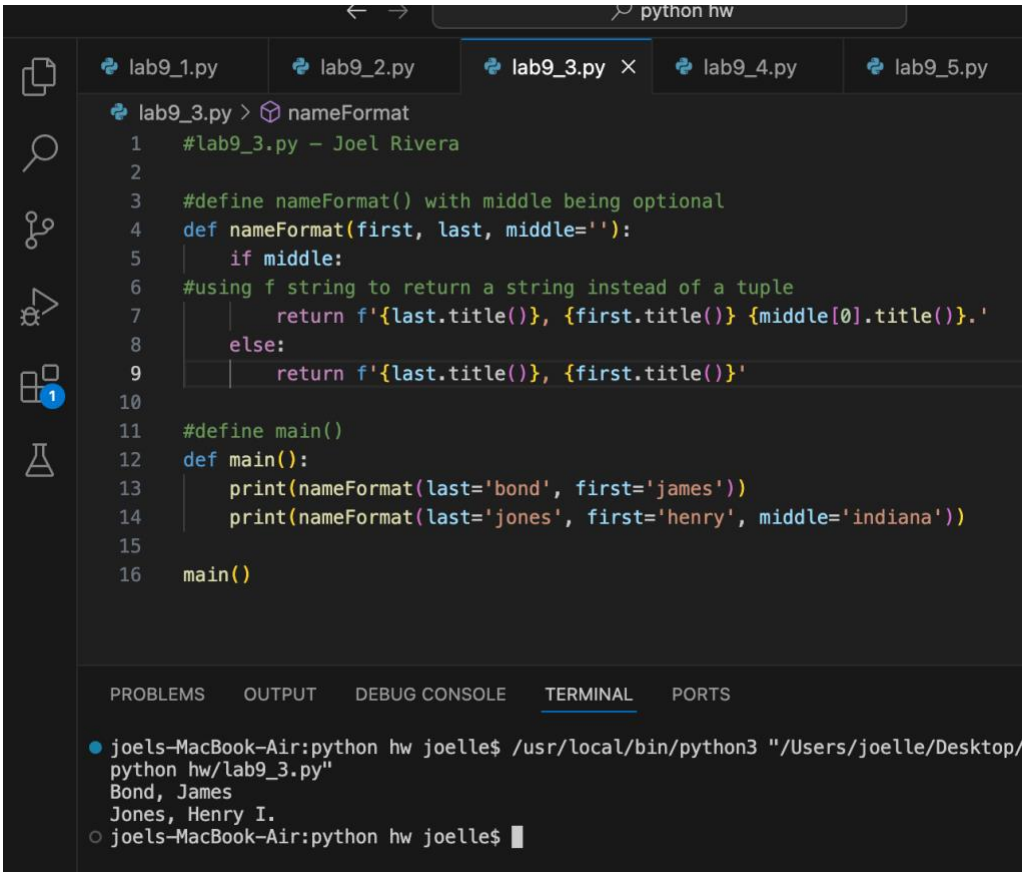
Et-574 F2F4L

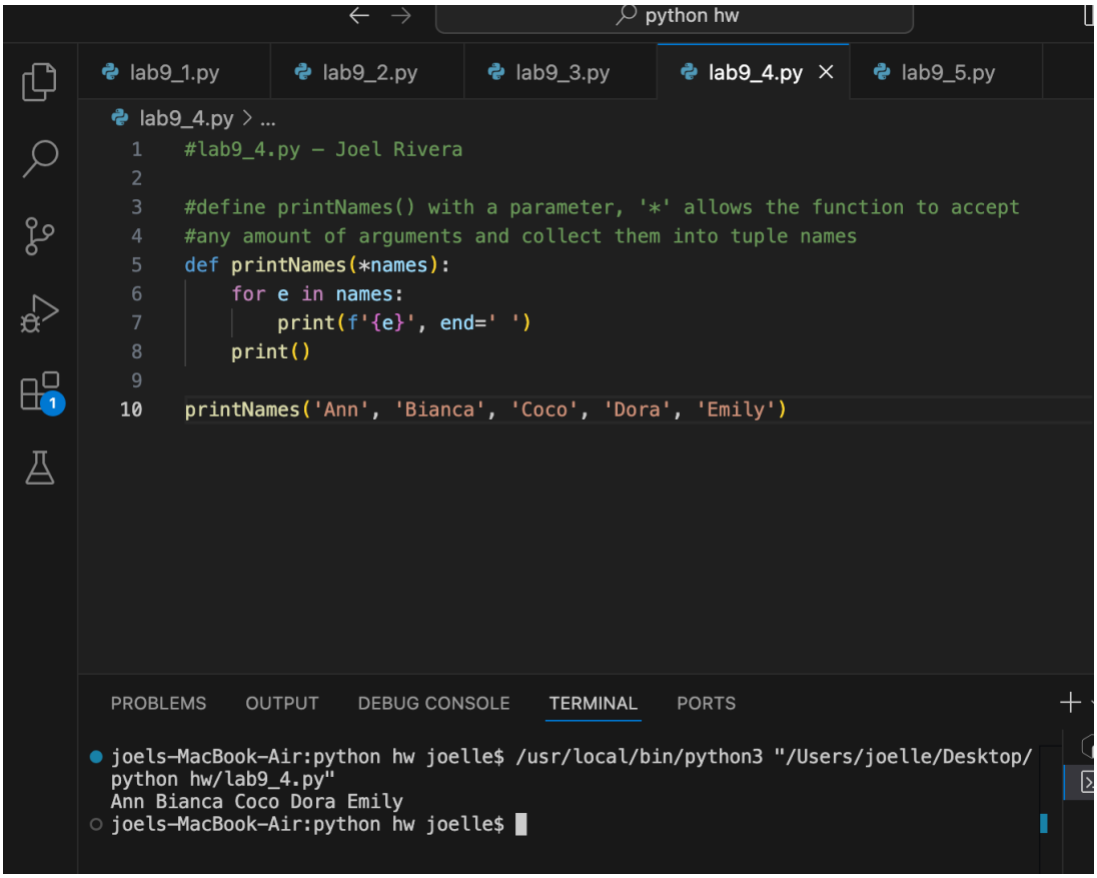
Lab 9

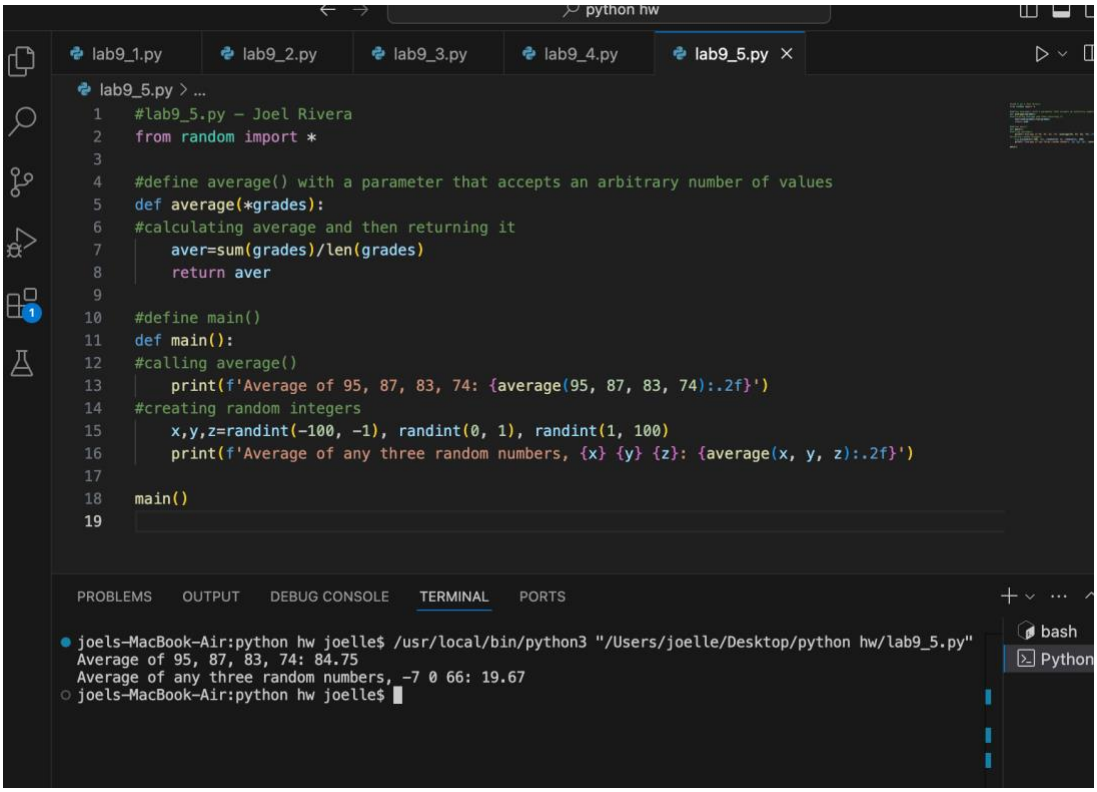
11/7/2024




lab9_1.py
lab9_2.py
lab9_3.py
lab9_4.py
lab9_5.py
lab9_2.py > ...
#lab9_2.py – Joel Rivera
#define function nameFormat()
def nameFormat(first, middle, last):
print(f'{first.title()} {middle[0].title()}. {last.title()}')
#define function main()
def main():
#calling nameFormat() with positional arguments
nameFormat('john', 'stu', 'smith')
#calling nameFormat() with keyword arguments
nameFormat(last = 'kennedy', first = 'john', middle = 'fitzgerald')
#calling main()
main()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_2.py"
John S. Smith
John F. Kennedy
joels-MacBook-Air:python hw joelle$


python hw
lab9_1.py
lab9_2.py
lab9_3.py
lab9_4.py
lab9_5.py
lab9_3.py > nameFormat
#lab9_3.py – Joel Rivera
#define nameFormat() with middle being optional
def nameFormat(first, last, middle=''):
if middle:
#using f string to return a string instead of a tuple
return f'{last.title()}, {first.title()} {middle[0].title()}.'
else:
return f'{last.title()}, {first.title()}'
#define main()
def main():
print(nameFormat(last='bond', first='james'))
print(nameFormat(last='jones', first='henry', middle='indiana'))
main()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_3.py"
Bond, James
Jones, Henry I.
joels-MacBook-Air:python hw joelle$


python hw
lab9_1.py
lab9_2.py
lab9_3.py
lab9_4.py
lab9_5.py
lab9_4.py > ...
#lab9_4.py – Joel Rivera
#define printNames() with a parameter, '*' allows the function to accept
#any amount of arguments and collect them into tuple names
def printNames(*names):
for e in names:
print(f'{e}', end=' ')
print()
printNames('Ann', 'Bianca', 'Coco', 'Dora', 'Emily')
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_4.py"
Ann Bianca Coco Dora Emily
joels-MacBook-Air:python hw joelle$


python hw
lab9_1.py
lab9_2.py
lab9_3.py
lab9_4.py
lab9_5.py
lab9_5.py > ...
#lab9_5.py – Joel Rivera
from random import *
#define average() with a parameter that accepts an arbitrary number of values
def average(*grades):
#calculating average and then returning it
aver=sum(grades)/len(grades)
return aver
#define main()
def main():
#calling average()
print(f'Average of 95, 87, 83, 74: {average(95, 87, 83, 74):.2f}')
#creating random integers
x,y,z=randint(-100, -1), randint(0, 1), randint(1, 100)
print(f'Average of any three random numbers, {x} {y} {z}: {average(x, y, z):.2f}')
main()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab9_5.py"
Average of 95, 87, 83, 74: 84.75
Average of any three random numbers, -7 0 66: 19.67
joels-MacBook-Air:python hw joelle$
bash
Python

6.I faced two challenges in this week's lab, the first was in question 3. I was trying to find a way to check if the user had inputted a 'middle' argument. Originally, I had written my if else statements in reverse, my old if statement checked if the user had inputted all three arguments and the output was wrong because I think the default value made the condition true always so even if there wasn't a 'middle' inputted, it was there anyway. When I made the change to only check if there was a 'middle', I then got the desired output. The second challenge I faced came from the 4th question, I had originally written a tuple inside of the function I was calling, but that wasn't the instruction. I reviewed the PowerPoint and saw that we had to use a '*' Infront of the parameter so the function is then able to accept any number of positional arguments and add those arguments into a tuple.