**COMP130 HW5: Software techniques
instructor: John MacCormick**

Question 1. (10 points) Consider the following Python code, which operates on a string variable s:

if s.startswith('a'):

 print('starts with "a"')

elif s.endswith('z'):

 print('ends with "z"')

Encapsulate this code into a function called check\_a\_z that takes s as a parameter.

Question 2. (10 points) The following code converts the Brazilian real currency to the Argentine peso currency, using the exchange rate of 255.7, which was the correct exchange rate on September 30, 2025.

def real\_to\_peso(x):

 pesos = x \* 255.7

 print(x, 'Brazilian reals equals', pesos, 'Argentine pesos')

Generalize this function by adding an additional parameter containing the exchange rate to be used.

Question 3. (15 points) The following two functions contain some repeated code.

def grams\_to\_kg\_and\_pounds(grams):

 kg = grams / 1000

 pounds = grams \* 0.00220462

 print(grams, 'grams is', kg, 'kilograms')

 print(grams, 'grams is', pounds, 'pounds')

def grams\_to\_ounces\_and\_pounds(grams):

 ounces = grams \* 0.035274

 pounds = grams \* 0.00220462

 print(grams, 'grams is', ounces, 'ounces')

 print(grams, 'grams is', pounds, 'pounds')

Define a new function that factors out the repeated code. Then rewrite the two functions above, eliminating the repeated code.

Question 4. (15 points) Suppose we need to add to the code from the previous question another function that works in a similar fashion. It has the signature grams\_to\_kg\_and\_ounces(grams), and it should print out results in kilograms and ounces. Write this new function, and re-examine all three grams\_to\_... functions. Now they contain more repeated code. Refactor and rewrite these functions to eliminate all repeated code, creating any additional functions as necessary.

Question 5. (10 points) Consider the following function, which can be used to format an amount of currency.

def print\_currency(currency, amount):

 print('The amount is ' + currency + '%.2f' % amount)

The expression '%.2f' % amount uses a technique we have not covered in class to format a numerical value to two decimal places. For example, using the standard currency abbreviation VND for the Vietnamese đồng, the invocation print\_currency('VND', 52.1) results in output The amount is VND52.10.

Add two guardians to this function. The first guardian should ensure that currency is a string. The second guardian should ensure that amount is nonnegative.

Question 6. (30 points) Based on the assigned reading and in-class discussion for our class on Algorithms and Ethics, write 200-300 words responding to the following prompt:

Describe and explain two sources of algorithmic bias in your own words.

Note: You are not expected to do additional research or use external sources other than the assigned reading when answering the above question. If you do use an external source, you must cite it. This includes generative AI programs.

Total points on assignment: 90