Jeffrey Olubajo

IT3883 W02

Due date: 5/3/2025

# Final Exam

## Sprint 1

### Requirements:

Story 1- A user would like to be able to calculate a combination of US coins to a total dollar amount. Example- 4 dimes and 7 quarters -> 2.15

Story 2- A user would want the calculation to be outputted as only a dollar amount with two decimal places.

### Design:

Pseudocode

Establish the coin values with their dollar value:

“penny”: 0.01, “nickel”: 0.05, “dime”: 0.10, “quarter”: 0.25

Make the sentence split with commas to get coin phrases [“1 penny”, “5 nickels”],

For the respective phrases:

* + Get the value and the coin name
  + Ensure the names are singular (pennies -> penny)
  + Calculating by getting product of number and the coin value
  + Put the result into the total

Round the total to two decimals.

### Implementation:

Source Code

def calc\_total(sentence):  
 coins = {  
 "penny": 0.01,  
 "nickel": 0.05,  
 "dime": 0.10,  
 "quarter": 0.25  
 }  
 # split the sentence phrases with commas  
 phrases = sentence.split(", ")  
 total = 0.0  
  
 for phrase in phrases:  
 parts = phrase.strip().split()  
 quantity = int(parts[0])  
 denom = parts[1].lower()  
  
 # Adjust plural denominations  
 if denom.endswith('ies'): # e.g., pennies  
 denom = denom[:-3] + 'y' # removes the last 3 characters  
 elif denom.endswith('s'): # e.g., nickels  
 denom = denom[:-1] # removes the last character  
 value = coins.get(denom, 0)  
 total += quantity \* value  
 return f"{total: .2f}" # two decimal places  
  
# Tests  
print(calc\_total("1 nickel and 17 quarters")) # output: 0.05 wrong  
print(calc\_total("21 pennies and 17 dimes and 52 quarters")) # output: 0.21 wrong  
print(calc\_total("1 dime and 1 nickel and 1 penny and 1 quarter")) # output: 0.10 wrong  
print(calc\_total("21 nickels and 15 pennies ")) # output: 1.05 wrong  
print(calc\_total("4 dimes and 7 quarters ")) # output: 0.40 wrong

I found that when my code was run, it did not account for the phrases after “and”, meaning I had to add an extra part of my code that splits the sentence with a comma when “and” is present so all values ae accounted for.

## Sprint 2

### Design:

Pseudocode

Establish the coin values with their dollar value:

“penny”: 0.01, “nickel”: 0.05, “dime”: 0.10, “quarter”: 0.25

Make the sentence split with commas to get coin phrases [“1 penny”, “5 nickels”]

\*\*Replace “and” with commas to split the phrases better.

For the respective phrases:

* + Get the value and the coin name
  + Ensure the names are singular (pennies -> penny)
  + Calculating by getting product of number and the coin value
  + Put the result into the total

Round the total to two decimals.

### Implementation:

Source Code

def calc\_total(sentence):  
 coins = {  
 "penny": 0.01,  
 "nickel": 0.05,  
 "dime": 0.10,  
 "quarter": 0.25  
 }  
 # Normalize sentence and split by 'and'  
 sentence = sentence.replace(" and ", ", ")  
 phrases = sentence.split(", ")  
 total = 0.0  
  
 for phrase in phrases:  
 parts = phrase.strip().split()  
 quantity = int(parts[0])  
 denom = parts[1].lower()  
  
 # Normalize plural denominations  
 if denom.endswith('ies'): # e.g., pennies  
 denom = denom[:-3] + 'y' # removes the last 3 characters  
 elif denom.endswith('s'): # e.g., nickels  
 denom = denom[:-1] # removes the last character  
 value = coins.get(denom, 0)  
 total += quantity \* value  
 return f"{total: .2f}"  
  
# Tests  
print(calc\_total("1 nickel and 17 quarters")) # output: 4.30 correct  
print(calc\_total("21 pennies and 17 dimes and 52 quarters")) # output: 14.91 correct  
print(calc\_total("1 dime and 1 nickel and 1 penny and 1 quarter")) # output: 0.41 correct  
print(calc\_total("21 nickels and 15 pennies ")) # output: 1.2 correct  
print(calc\_total("4 dimes and 7 quarters ")) # output: 2.15 correct