**Subroutines: Functions and Procedures**

1. **Program 1**

'''

Converting currenices

Miles and Kilometers

Kilograms and Pounds

Celsius and Fahrenheit

'''

def print\_menu():

print('1. British Pounds to US Dollars')

print('2. British Pounds to Euros')

print('3. British Pounds to Yens')

def Pounds\_Dollars():

Pounds = float(input('Enter Pounds: '))

Dollars = Pounds \* 1.28

print('Amount in Dollars is: ',round(Dollars,2))

def Pounds\_Euros():

Pounds = float(input('Enter Pounds: '))

Euros = Pounds \* 1.19

print('Amount in Euros is: ',round(Euros,2))

def Pounds\_Yens():

Pounds = float(input('Enter Pounds: '))

Yen = Pounds \* 139.79

print('Amount in Yens is:',round(Yen,2))

def main():

print\_menu()

choice = input('Which conversion would you like to do? ')

if choice == '1':

Pounds\_Dollars()

if choice == '2':

Pounds\_Euros()

if choice == '3':

Pounds\_Yens()

main()

1. **Program 2**

def get\_formatted\_name(first\_name, last\_name, middle\_name=''):

"""Return a full name, neatly formatted."""

if middle\_name:

full\_name = first\_name + ' ' + middle\_name + ' ' + last\_name

else:

full\_name = first\_name + ' ' + last\_name

return full\_name.title()

musician = get\_formatted\_name('jimi', 'hendrix')

print(musician)

musician = get\_formatted\_name('john', 'hooker', 'lee')

print(musician)

1. **Program 3**

def describe\_pet(pet\_name, animal\_type='dog'):

"""Display information about a pet."""

print("\nI have a " + animal\_type + ".")

print("My " + animal\_type + "'s name is " + pet\_name.title() + ".")

# A dog named Willie.

describe\_pet('willie')

describe\_pet(pet\_name='willie')

# A hamster named Harry.

describe\_pet('harry', 'hamster')

describe\_pet(pet\_name='harry', animal\_type='hamster')

describe\_pet(animal\_type='hamster', pet\_name='harry')

1. **Program 4 – call it pizza.py**

def make\_pizza(size, \*toppings):

"""Summarize the pizza we are about to make."""

print("\nMaking a " + str(size) +

"-inch pizza with the following toppings:")

for topping in toppings:

print("- " + topping)

make\_pizza(16, 'pepperoni')

make\_pizza(12, 'mushrooms', 'green peppers', 'extra cheese')

1. **Program 5 (related to Program 4)**

import pizza as p

p.make\_pizza(16, 'pepperoni')

p.make\_pizza(12, 'mushrooms', 'green peppers', 'extra cheese')

1. **Program 6**

def build\_person(first\_name, last\_name, age=''):

"""**Return a dictionary of information about a person."""**

person = {'first': first\_name, 'last': last\_name}

if age:

person['age'] = age

return person

musician = build\_person('jimi', 'hendrix', age=27)

print(musician)

1. **Program 7**

def greet\_users(names):

"""Print a simple greeting to each user in the list."""

for name in names:

msg = "Hello, " + name.title() + "!"

print(msg)

usernames = ['hannah', 'ty', 'margot']

greet\_users(usernames)

1. **Program 8**

def greet\_user(username):

"""Display a simple greeting."""

print("Hello, " + username.title() + "!")

greet\_user('jesse')

1. **Program 9**

def print\_models(unprinted\_designs, completed\_models):

"""

Simulate printing each design, until there are none left.

Move each design to completed\_models after printing.

"""

while unprinted\_designs:

current\_design = unprinted\_designs.pop()

# Simulate creating a 3d print from the design.

print("Printing model: " + current\_design)

completed\_models.append(current\_design)

def show\_completed\_models(completed\_models):

"""Show all the models that were printed."""

print("\nThe following models have been printed:")

for completed\_model in completed\_models:

print(completed\_model)

unprinted\_designs = ['iphone case', 'robot pendant', 'dodecahedron']

completed\_models = []

print\_models(unprinted\_designs, completed\_models)

show\_completed\_models(completed\_models)