Objectives:

* Creating functions.
* Setting and passing arguments.
* Creating global variables.

**There are 5 challenge exercises with 5 print screens, each worth 20%**

Please submit this document for grading when completed… Please work in groups.

A function is a name that you can call, and within a function are code statements. It’s a neat way to break down your code.

**Project #1** (here we create two functions message1 and message2). We use the keyword **def** to use a function.

Graphical user interface, text, application

Description automatically generated

**Project #2** (using input and output in a function)

Graphical user interface, text, application

Description automatically generated

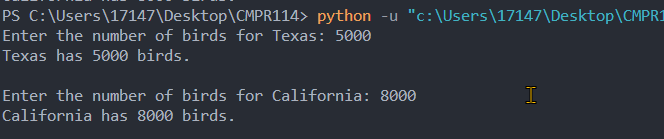
**Project #3** (calling the first function and in between functions)

Text

Description automatically generated

**Challenge Exercise #1**: using project #3, add to the code so that the user can enter or input the number of birds in each state.

**#1 Print screen the output with the code below here.**



Code:

for i in ['Texas', 'California']:

birds = int(input(f"Enter the number of birds for {i}: " ))

print(f"{i} has {birds} birds.\n")

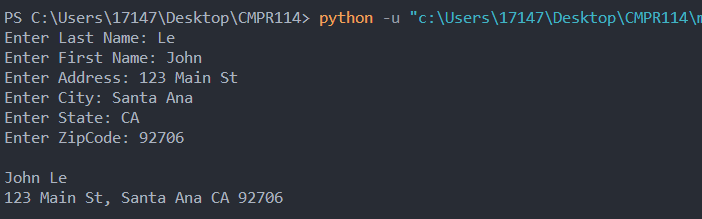
**Project #4** (passing string arguments to a function).

Text

Description automatically generated

**Challenge Exercise #2:** Create a program with a *function* that will allow the user to enter the last, and first names, addresses, city, and state, with zip codes.

**#2 Print screen the output with the code below here.**



Code:

def get\_fullname():

lastname = input("Enter Last Name: ")

firstname = input("Enter First Name: ")

fullname = firstname + " " + lastname

return fullname

def get\_address():

address = input("Enter Address: ")

return address

def get\_city():

city = input("Enter City: ")

return city

def get\_state():

state = input("Enter State: ")

return state

def get\_zipcode():

zipcode = input("Enter ZipCode: ")

return zipcode

name = get\_fullname()

address = get\_address()

city = get\_city()

state = get\_state()

zip = get\_zipcode()

print("\n" + name)

print(address + ", " + city + " " + state + " " + zip + "\n")

**Project #5** (Creating a global variable)

A global variable is used using the global keyword, and it’s a variable that can be seen anywhere on the program. See the code below.

Graphical user interface, text, application

Description automatically generated

**Project #5 continued**, using global variables, this example will add 3+4 using a function named add.

Text

Description automatically generated

Now, cut and paste print(total) below add (3,4). Notice the error appears because it is not a part of the function no more.

Text, letter

Description automatically generated

To fix this, you can use the **return** word in this local variable and assign a local variable of (a) and print (a).

Text, letter

Description automatically generated

Now, let’s talk about a global variable. To use a **global** variable, use the global keyword. Notice, that we the total global variable can be seen anywhere in the program, since its global.

Text

Description automatically generated

**Challenge Exercise #3:** Modify project #5 to add three numbers.

**#3 Print screen the output with the code below here.**



Code:

def add(num1, num2, num3):

global total

total = num1 + num2 + num3

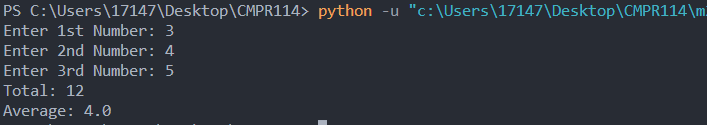
return total

add(3,4,5)

print(f"Total : {total}")

**Challenge Exercise #4:** Modify project #5 so the user can enter or input any three numbers, be sure to use global variables in the program. Also, sum and average the three numbers.

**#4 Print screen the output with the code below here.**



Code:

def add():

global total

total = num1 + num2 + num3

return total

global num1, num2, num3, total

num1 = int(input("Enter 1st Number: "))

num2 = int(input("Enter 2nd Number: "))

num3 = int(input("Enter 3rd Number: "))

add()

avg = total / 3

print(f"Total: {total}")

print(f"Average: {avg}")

**Project #6** (using a global constant, which means a global variable where a parameter or a number is assigned).

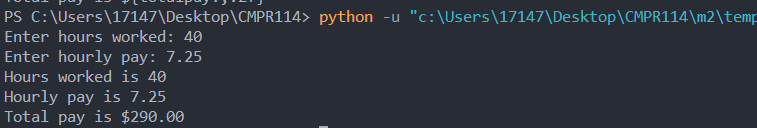
Text

Description automatically generatedText

Description automatically generated

**Challenge Exercise #5:** using a **function** and **setting** and **passing** **arguments**, create a program that will ask the user to enter the hours worked and hourly pay. Then get the output in a print statement.

**#5 Print screen the output with the code below here.**

****

Code:

def calc\_pay(hours\_worked, hourly\_pay):

totalpay = hours\_worked \* hourly\_pay

return totalpay

hours\_worked = int(input("Enter hours worked: "))

hourly\_pay = float(input("Enter hourly pay: "))

totalpay = calc\_pay(hours\_worked, hourly\_pay

)

print(f"Hours worked is {hours\_worked}")

print(f"Hourly pay is {hourly\_pay}")

print(f"Total pay is ${totalpay:,.2f}")

**Submit this document to the Module 2 Class Exercise.**