**Mission 8.6: Importing Custom Functions and VariableS**

I. Situation:

1. Importing functions and variables
   1. Up until this point, your programs have been composed of just one file. Although we have been calling them “programs,” they are more often referred to as “scripts.” In the common language of programming, a “program” usually consists of several “scripts” that work together.
   2. Programming is often done in what is called a “modular” way, where different pieces of code are developed independently and then put together when the program runs. In this mission, you will make your first modular program.
2. An Example of Modular Programming
   1. Imagine a game where the user can choose to play as different characters. Some characters are faster, some are bigger, and maybe some are tougher. When a user selects a character to use, the program will need to use the variable values for that character but doesn’t need the information for the other characters.
   2. The **import** function allows a script to use a function or variable that is defined in a different .py file IN THE SAME DIRECTORY as it is. To import a .py file, simply use the command *import filename*, where filename is the file you wish to access. Notice that the .py file extension is not necessary.
   3. Consider the following example:

The directory your program runs in has 4 files in it. They are named main.py, zarg.py, zeleste.py, and zifnab.py.

*main.py:*

choice = input(“Which character do you want to play, zarg, zeleste, or zifnab?”)

if choice == “zarg”:

import zarg

elif choice == “zeleste”:

import zeleste

elif choice == “zifnab”:

import zifnab

print (strength, speed, magic)

*zarg.py: zeleste.py: zifnab.py:*

strength = 10 strength = 3 strength = 3

speed = 3 speed = 10 speed = 3

magic = 0 magic = 3 magic = 10

As you can see, this is an easy way to import whole sets of variables and / or functions at one time. In this example, only variables were imported, but this works the same way for functions too.

* 1. You can also use what is called a “function call” to use functions that you have imported. For example, consider this example:

There are 3 files in the program directory: main.py, function1.py, and function2.py

*main.py*

import function1

import function2

select = input("Would you like function 1, function 2, or exit?")

if select == "1":

function1.func1 () ### *This is the function call for func1 in the file function1.py imported a few lines above*

elif select == "2":

function2.func2 () ### *This is the function call for func2 in the file function2.py imported a few lines above*

elif select == "exit":

print ("thank you, goodbye.")

*function1.py function2.py*

def func1 (): def func2 ():

print ("This is function 1.") print ("This is function 2.")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Notice that in this case, the function is imported, and then “called” from the main script.

As you can see in the example above, the syntax for a function call is *filename.functionname ()*

* 1. There are LOTS of custom modules out there that include functions and variable values that make programming in Python VERY versatile and powerful. You can see a list of 25 common modules at <https://wiki.python.org/moin/UsefulModules> if you are interested. They include things like PyGame, for developing games, and everything else you could imagine. The online site <https://www.pypi.org> currently lists nearly 1.2 MILLION module releases that support everything from 3D geometry to computational molecular biology, and everything in between.

**II. Mission: Your team will write a script that allows you to run all of the programs you have written so far by using function calls.**

1. Save a version of your existing programs as function scripts:
   1. Make copies of your existing .py files in drive, and rename them *xxxxxfirstprogramfunc.py*, *xxxxxmission3func.py*, and *xxxxxsimplecalcfunc.py*.
   2. In each of those scripts, add a line of code at top that defines the whole script as a function. Name the function something that makes sense.
2. Write a xxxxxmain.py script that uses function calls to access your three previous programs:
   1. At the top of the script, import your three programs.
      1. EXAMPLE: import 80513firstprogramfunc.firstprogram ()
   2. Next, code an input menu that gives the user the options “first program”, “inputs and outputs”, “simple calculator”, and “exit”.
   3. Use if/elif/else to code function calls that access the appropriate functions
3. Add a loop by making the main script a function that can call itself.
   1. This is just like what you did in mission 5 to loop the calculator program back to a menu of options
4. Test and debug your program
   1. Remember, for all of this to work the .py files need to be in the same directory. This means you will have to save them to the hard drive using XFCE. Make a folder named xxxxxpythonportfolio in home/chrome to save them