**Virtual Initio Programming: Modules**

**AIM:** After completing this worksheet you should be able to explain what a namespace is in programming and use create and use modules in Python.

**You Need:** To complete this worksheet you need to have a virtual Initio simulator (see WS1), and to be able to use files to store Programs (WS5). You also need to know the commands to operate the Initio motors and sensors (WS3 & WS4). You should be able to use If statements (WS7), while loops (WS8 & WS10), variables (WS12), data types (WS14) and Functions (WS16) in Python programs.

**If the simulator isn’t already running: Start the Simulator, Select the Initio Simulation and house.xml, then start IDLE (open a *new IDLE window* if you have used IDLE to start the simulator).**

You have already met the **import** keyword at the start of your programs which you have been using to import useful functions into your programs. These functions are stored in *modules* such as the **time** module.

You can create your own modules by storing all your functions in a file.

Create a file called turning.py this should contain the functions from WS16: turn(side) and obstacle(side)

You will need to import simclient.simrobot as initio and time at the start of this module as usual.

Type the following at the command line:

>>> import turning

>>> turning.initio.init()

>>> turning.turn('left')

>>> turning.initio.stop()

**Question 1:** What happens?

**Notice** that you have to use the file name turning at the start of all the functions you want to use in the file.

You have created a *module* which has the *namespace* turning.

If you want to change the namespace then you can use the import … as … syntax, like you have been doing for import simclient.simrobot as initio

**Important:** Type

>>> turning.initio.cleanup()

Before you import the module again for the next question.

**Question 2:** What sequence of commands would you need to type to import turning using the namespace my\_turning and then turn the virtual Initio to the right?

**Reloading a Module**

Although you can give a module a new namespace by reloading it. The Python command line only loads modules once. So, if you edit your module and try importing again it won’t use your changes.

**Exercise 1:** Edit your module so that it prints out a message when it executes the **turn** function. Then import it again and try executing turn.

What happens and why?

In order to properly reload the module you need to use something called **importlib.** Type the following:

>>> import importlib

>>> importlib.reload(turning)

Now try executing your new version of the turn function.

**Question 3:** What happens?

**Running code when you import a module.**

Suppose you wanted to initialise the Initio whenever you loaded the module so you didn’t have to type turning.initio.init() at the Python command line? This is easy, you just include the command in the module – just as in a normal program file.

**Exercise 2:** Adapt your module so that that it initialises the Initio when it is imported.



University of Liverpool, 2019

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).