EMAT10007 – Introduction to Computer Programming

Exercises – Week 6. Importing Python Files

6.1 Packages and Modules

Essential Questions

Exercise 1 - Importing a Python Module

```
geometry/
|
|---- __init__.py
|---- main.py
|---- volumes.py
```

1. Create the file system shown above. Leave all .py files empty to begin with.
2. In file volumes.py, write two functions, `sphere` and `cube`. Each function should take one input argument and return the volume of the shape in the name of the function. Units of the output should match the input (i.e. input in cm, output in cm^3).
3. Edit the content of main.py so that when it is run, it prints the volume of a sphere of radius 3 m.
4. Is there a way to achieve the same operation, using shorter code in main.py?.
5. Add another file, areas.py within the ‘geometry’ folder. In areas.py, write two functions, `sphere` and `cube`. Each function should take one input argument and return the surface area of the shape in the name of the function. Units of the output should match the input (i.e. input in cm, output in cm^2).
6. Edit the content of main.py so that a variable `A = 5 m` is created and the area and volume of a sphere of radius `A` and a cube of side length `A = 5 m` are printed.

Advanced Questions

(A) Add your another .py file to the directory and create a Python module of your choosing (it does not need to be related to geometry). For example, you could store some useful equations that you used recently in another unit as Python functions. Or you could simply take some of the functions you write in Week 4 and practise assembling them as a Python module and importing them.

(B) Practise different ways of calling variables, functions and classes from within main.py such as changing the namespace and importing individual functions and/or variables.

6.2 Importing from other locations

Essential Questions

```
cubes/
|
|---- __init__.py
|---- cube.py
|---- shapes/
        |
        |---- main.py
        |---- spheres/
                |
                |---- __init__.py
                |---- sphere.py
```

Create the file system shown above. Leave all .py files empty to begin with.

Exercise 2 - Importing from downstream locations

1. In file sphere.py, create a class `Sphere` that takes one input argument, the radius.
2. Give the class two methods that return the surface area and volume of the sphere respectively.
3. Edit main.py to print the surface area of a sphere of radius 5 m.
   Hint: There are a number of ways to deal with the need for constants e.g. pi
   The most reliable way to ensure a constant value is used is to import it from:
   - an external package e.g. `math` (for well known constants like pi)
   - a module created by you to store for program-specific constants - we'll try this out in the next question...
4. Create a sub-directory within 'spheres' and a file within it, dimensions.py.
5. Create a variable within dimensions.py, `radius` and give it a numerical value.
6. Import the entire contents of dimensions.py to main.py and use `radius` as the input argument to generate a `Sphere` object.

Exercise 3 - Importing from upstream locations

1. In file cube.py, create a class `Cube` that takes one input argument, the side length with a default value of 1 m.
2. Give the class two methods that return the surface area and volume of the cube respectively.
3. Edit main.py to print the surface area of a cube of side 10 cm.

Advanced Questions

(A) Create another sub-directory within the ‘shapes’ sub-directory and create your own python package. Create Python file(s) within the directory and practise calling them from within main.py