Lab title: SOAP webservices.
Learning outcomes: learn to create a SOAP webservice using Python.

Install olapy, spyne and suds-community libraries in pipenv using the following commands:
pipenv install olapy
pipenv install spyne
pipenv install suds-community

1. Suds SOAP client
In this part we will use some public web services.

The following url contains the WSDL for a SOAP mathematical service. Enter this url into any web browser to view the WSDL file in question.
http://www.dneonline.com/calculator.asmx?WSDL

Suds is a python SOAP client which allows one to interact with existing SOAP web services. Suds has been installed on the machines in this lab. The following sample Python code uses a SOAP web service to add two numbers.

```
from suds.client import Client
url = "http://calculator-webservice.mybluemix.net/calculator?wsdl"
client = Client(url)
print(client) # This returns the details regarding the webservice.
print(client.service.Add(2,3))
```

suds_client.py

Do a web search to determine what other SOAP web services are available.

2. Spyne SOAP server
In this part we will create a SOAP server. This will be achieved using spyne which is a python framework for creating webservices.

Briefly examine the documentation for spyne located at the following address
http://spyne.io/docs/2.10/index.html

The following code creates a SOAP webserver (see helloworld_soap.py which can be downloaded from learning central). This code has been adopted from the following URL. Briefly examine the documentation corresponding to this example to gain a better understanding of the code.
http://spyne.io/docs/2.10/manual/02_helloworld.html

```
from spyne import Application, rpc, ServiceBase, Iterable, Integer, Unicode
from spyne.protocol.soap import Soap11
from spyne.server.wsgi import WsgiApplication

class HelloWorldService(ServiceBase):
    @srpc(Unicode, Integer, _returns=Iterable(Unicode))
```

```
    def say_hello(name, times):
        l = []
        for i in range(times):
            l.append(name)
        return l

application = Application([HelloWorldService], 'spyne.examples.hello.soap',
                          in_protocol=Soap11(validator='lxml'),
                          out_protocol=Soap11())
wsgi_application = WsgiApplication(application)

if __name__ == '__main__':
    from wsgiref.simple_server import make_server
    server = make_server('127.0.0.1', 8000, wsgi_application)
    server.serve_forever()
```

helloworld_soap.py

Open a terminal and run the above script. This will start the SOAP server. Visit the following URL to view the WSDL file which has been created:
http://127.0.0.1:8000/?wsdl

3. Suds SOAP client

The following code contains a SOAP client which tests the above web service:

```
from suds.client import Client
hello_client =Client('http://localhost:8000/?wsdl')
print(hello_client.service.say_hello("Dave", 5))
```

helloworld_suds.py

Create a new SOAP web service which takes two numbers and returns the corresponding sum.
Create a new SOAP web service which returns the number of times that webservice has been called. (hint: create a global variable in main)
Create a web service which integrates two existing web services.
Create a web service which accesses the data stored in a remote object.