## Summary

This task focuses on the design of a client-server network that can be used to send basic data across. The programs should be able to serialize structured data and sent it over the network. The client should be able to read text files and send the content to the server.

The structured dictionary should be serialized into multiple formats, namely XML, JSON and Pickle. Pickle is a python library that serializes a python object into a stream of bytes that can be repackaged on the receiving end to form the original object.

The server should be able to print the contents received onto the screen or a file. In addition, the server should be able to handle encrypted contents.

The language used is python version 3.8 and the code is written to standard with elaborate unit tests. The write below focuses on the unit tests only.

# The client - Unit tests

Unit tests are written in python using the **unittest** module that’s provided natively. Writing tests is mainly an assertion-oriented practice where the testing code attempts to predict the correct outcome of a functions execution and asserts a logical statement to that data.

##### What’s tested

For the client, tests are created for most of not all the modular bits in the code. In other words, every function, save for the main one are tested (the main function contains the main application stream – starts the client app)

###### Testing server active

This test is aimed to test whether the server is active; it does so by attempting to connect. If the connection is successful, it asserts true, else if an error occurs, a false is asserted.

###### Test JSON serialization

Done simply by asserting the output of the serialize function on the client with the JSON option.

Making sure this **{"a": 1, "b": 2}** outputs

This **b‘{\n    "a": 1,\n    "b": 2\n}’**

###### Test XML serialization

Similar to testing JSON but a different output is asserted – an XML string

###### Test Pickle serialization

Matching the stream HEX bytes generated by the serialization, to the expected value if Pickle were to be used directly.

Other tested methods include encryption which basically asserts true if no error occurs in the encryption process, passing final data which tests the respective function asserting the output to the expected string and also the send data function is tested. Worth noting that if the test server active test fails then the send data test inevitably fails as well.

# The Server - Unit tests

Similar to the client, all but the main modular bits are tested, which goes to show why functional code is much easier to test than monolithic code.

##### Summary of Server tests

Tests are written to test decryption; this is done by using the encrypt function from the client and deciphering the output on the server using the decrypt function and asserting equality to the original string sequence.

File creation is tested where we assert true to the return value of the file creation function in the server.

Also tested is the xml de-serialize function in which an XML string is expected to be converted to a regular dictionary object.