Python for Tool Developers

# Lab 6

# 11/15/2016

### Lab 6. Network Programming with Unicode strings

As we've demonstrated, Python handles unicode to byte string conversions very differently in Python 2 and Python 3. In the data directory is a naive Python 3 client server application consisting of a *client.py* file that sends unicode strings and a *server.py* file that receives them and prints them out. However, as it stands, this application doesn't work. There are multiple issues regarding unicode to byte string conversion on both the client and the server.

1. Fix the all problems with this application so that the server will print out the unicode string without throwing any Unicode encoding or decoding exceptions.

2. Convert the client and server to use Python 2. If you wish to keep the context manager, you can import the closing function from the contextlib library in Python 2 like so:

|  |
| --- |
| from contextlib import closing  with closing(socket.socket(socket.AF\_INET,socket.socket.D\_GRAM)) as sock |

Otherwise, wrap your socket functions in a try/except/finally block.

3. Attempt to run the client and the server. What is the difference in the behavior from Python 2 and Python 3?