Lab 2 Questions

By Sandy Huang

**Question 1**: How do you specify a TCP socket in Python?

To specify a **TCP socket** in Python first you must use the **“socket”** library. By invoking the method “**socket”** the “TCP socket” is created and hosted on **“AF\_INET”** (providing the hostname or IPv4 address) and the “SOCK\_STREAM” which specifies the socket type.

**Question 2**: What is the difference between a client socket and a server socket in Python?

**A “client socket” initiates a connection with the server to receive/send data.**

**As “server socket” listens and waits to accept connections, receives data from a “client socket” then waits and sends back data until the connection is closed.**

**Question 3**: How do we instruct the OS to let us reuse the same bind port?

**In “echo\_server.py”, we are given the method “setsocketopt()” which has one of its arguments as “socket.SO\_REUSEADDR”. “socket.SO\_REUSEADDR” tells the OS that the local socket is to be reused without waiting for its natural timeout to expire.**

**Question 4**: What information do we get about incoming connections?

**When we receive an incoming connection we receive information about the address bound to the host and port connections where host consists of either an IPv4 or a DNS and port is an integer.**

**Question 5**: What is returned by recv() from the server after it is done sending the HTTP request?

**The method ‘recv()’ returns a bytes object representing the data received whenever a message is sent to the server. It takes in “bufsize” as a parameter to specify the maximum of bytes it can read (or length of message that can be received) in our case 1024.**

**The example in ‘Foobar’ returns b'Foobar\n'**

**Question 6**: Provide a link to your code on GitHub.

<https://github.com/shearpaladin/CMPUT404-LAB2>