William Kenji Kiplinger
https://github.com/Kenjum/CS380-EX3

```
_______________________________Ex3Client.java_____________________________________
import java.net.Socket;
import java.nio.ByteBuffer;
import java.io.InputStream;
import java.io.OutputStream;
import javax.xml.bind.DatatypeConverter;

public final class Ex3Client {
        public static void main(String[] args) throws Exception {
                try (Socket socket = new Socket("18.221.102.182", 38103)) {

                        System.out.println("Connected to server.");

                        InputStream is = socket.getInputStream();
                        OutputStream os = socket.getOutputStream();

                        // First value is how many values we will receive. We save that
                        // in everything.
                        int sizeFromServer = is.read();
                        System.out.println("Reading " + sizeFromServer + " bytes.");

                        // Stores the following values in a array the size of sizeFromServer
                        byte[] message = new byte[sizeFromServer];
                        for (int i = 0; i < message.length; i++) {
                                message[i] = (byte) is.read();
                        }

                        // We get what is in message and convert that to hex to
                        // messsageHex.
                        String messageHex = DatatypeConverter.printHexBinary(message);
                        System.out.print("Data received:\n   ");

                        // When the message reaches 20 characters, it will send it to the
                        // next line. This is going 1 character at a time.
                        int newLine = 0;
                        for (int i = 0; i < messageHex.length(); i++) {
                                if (newLine == 20) {
                                        System.out.println();
                                        System.out.print("   ");
                                        newLine = 0;
                                }
                                System.out.print(messageHex.substring(i, i + 1));
                                newLine++;
                        }
```

```
            // send to checksum method. crcCheckByte uses
            // ByteBuffer.allocate(4).putInt(crcCheck).array();
            // to allow storage of a buffer in a byte array 4 bytes large. It's
            // then converted to Hex so it can be properly output.
            short crcChecksum = checksum(message);
            byte[] crcCheckByte = ByteBuffer.allocate(4).putShort(crcChecksum).array();
            String crcString = DatatypeConverter.printHexBinary(crcCheckByte);
            System.out.println("\nChecksum calculated: 0x" + crcString.substring(0, 4) + ".");

            // send what we have to the server and get a response to see if it
            // is correct.
            os.write(crcCheckByte);
            int serverResponse = is.read();
            if (serverResponse == 1)
                System.out.println("Response good.");
            else
                System.out.println("Response bad.");

            System.out.println("Disconnected from server.");
        }
    }

    public static short checksum(byte[] b) {
        long sum = 0;
        int index = 0;

        // This is a conversion of what was shown in the exercise in the C
        // language.
        for (int i = b.length; i > 0; i--) {
            sum += (b[index++] & 0xFF) << 8;
            if ((--i) == 0)
                break;
            sum += (b[index++] & 0xff);
        }
        return (short) ((~((sum & 0xFFFF) + (sum >> 16))) & 0xFFFF);
    }
}
```