Christopher Rivas
https://github.com/Rivas1/CS-380-Exercise-3

---

```
import java.io.*;
import java.net.*;


public final class Ex3Client
{
	public static void main (String[] args) throws IOException
	{
		// int numberOfBytes = -1; // stores number of bytes to be received [0,255]
		int n = -1;
		int g = 0;
		int[] bytesFromServer;
		short checkSum = -1;
		String cs = "";
		try
		{
			/* make a connection */
			Socket socket = new Socket ("codebank.xyz", 38103);
			if (socket.isConnected())
				System.out.println("Connected to server.");

			/* create stream from Socket */
			InputStream IS = socket.getInputStream();
			PrintStream PS = new PrintStream(socket.getOutputStream()); // out from client to server
			/* read in byte that contains number of bytes to be received */
			n = obtain(socket, IS);
			System.out.println("Reading " + n + " bytes.");

			/* If not between 0 and 255, quit. */
			if ( n > 255 || n < 0 )
			{
```

```
                System.out.println("Value is not between 0 and 255!\nProgram will now
terminate.");
                System.exit(0);
        }

        /* Read corresponding number of bytes into integer array */
        bytesFromServer = new int[n];
        read_in_bytes(bytesFromServer, socket, IS);

        /* Print data received */
        System.out.print("Data received: ");
        for ( int i = 0; i < bytesFromServer.length; i++ )
        {
                if ( (i % 10) == 0)
                        System.out.println();
                System.out.print( Integer.toHexString(bytesFromServer[i]) );
        }

        /* Calculate checksum */
        checkSum = checksum( bytesFromServer );

        /* Convert check sum to 2 bytes with padded zeros for 16 bits */
        cs = Integer.toHexString(checkSum & 0xffff);
        g = (4 - cs.length() );
        while ( g > 0 )
        {
                cs = "0" + cs;
                g--;
        }

        /* Send check sum value to server */
        PS.println(cs);
        if ( IS.read() == 0 )
```

```
                        System.out.println("Response is bad.");
                else if ( IS.read() == 1 )
                        System.out.println("Response is good.");
        }
        catch (IOException e)
        { e.printStackTrace(); }
    }
    public static int obtain ( Socket socket, InputStream IS ) throws IOException
    {
        try
        {
            return IS.read();
        }
        catch (IOException e)
        { e.printStackTrace(); }
        return IS.read();
    }
    public static void read_in_bytes ( int[] bytesFromServer, Socket socket, InputStream IS )  throws
IOException
    {
        try
        {
            for ( int i = 0; i < bytesFromServer.length; i++ )
                bytesFromServer[i] = IS.read();
        }
        catch ( IOException e )
        { e.printStackTrace(); }
    }
    public static short checksum( int[] b )
    {
        int x = -1;
        int sum = 0;
        String c = "";
```

```
String hex = "";
int a = 0;
String left16bits = "";
String right16bits = "";
int l = 0, r = 0;
int var1 = 0; // stores value obtained from adding left 16 bits to right 16 bits
String binary = "";
String inverted = "";
short checksum = 0;
for ( int i = 0; i < b.length-1; i = i + 2)
{
        // Concatenate 2 bytes
        c = Integer.toString( b[i] ) + Integer.toString( b[i+1] );
        // convert back to integer
        x = Integer.parseInt( c );
        // add to sum
        sum = sum + x;
}
hex = Integer.toHexString(sum);

/* pad hexadecimal string with zeros to ensure 32 bits */
a = 8 - hex.length();
while ( a > 0 )
{
        hex = "0" + hex;
        a--;
}

/* Add left 16 bits to right 16 bits */
left16bits = hex.substring(0,4);
right16bits = hex.substring(4,8);
l = Integer.parseInt(left16bits, 16);
r = Integer.parseInt(right16bits, 16);
```

```
            var1 = l + r;

            /* Calculate 1's complement for 1's complement sum:
              1. Write number as binary
              2. Flip all bits.
            */
            binary = Integer.toBinaryString(var1);
            inverted = binary.replaceAll("0", "x").replaceAll("1", "0").replaceAll("x", "1");

            /*Convert to short*/
            checksum = Short.parseShort( inverted, 2 );
            String hsum = Integer.toHexString( checksum & 0xffff);
            System.out.println("\nChecksum calculated: 0x" + hsum + ".");

            // temporary
            return checksum;
        }
}
```