/\*\*

\* Author: Colin Koo

\* Professor: Davarpanah

\* Assignment : Exercise 2: We are connecting to a hosted server, then receiving bytes from the server in segments.

\* We then put the bytes together and send a CRC32 key back to the server.

\*/

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Socket;

import java.net.UnknownHostException;

import java.util.zip.CRC32;

public class Ex2Client {

public static void main(String[] args) throws UnknownHostException, IOException {

try (Socket socket = new Socket("codebank.xyz", 38102)){

System.out.println("Connected to: " + socket.getInetAddress() + ":" + socket.getPort() + "\n");

OutputStream os = socket.getOutputStream();

InputStream is = socket.getInputStream();

byte fullMsg[] = new byte[100];

byte message1, message2;

/\*\*

\* Reads the half-bytes received from the server then concatenates them into a byte by shifting the first message then

\* using the OR bitwise operator.

\*/

for (int i = 0; i < 100; i++){

message1 = (byte) is.read();

message2 = (byte) is.read();

message1 <<= 4;

fullMsg[i] = message1;

fullMsg[i] = (byte) (fullMsg[i] | message2);

}

System.out.print("Received bytes: ");

for (int i = 0; i < fullMsg.length; ++i){

System.out.print(Integer.toHexString(fullMsg[i] & 0xFF).toUpperCase());

}

CRC32 crc = new CRC32();

crc.update(fullMsg);

long crcResult = crc.getValue();

System.out.println("\n\nGenerated CRC32: " + Integer.toHexString((int) crcResult).toUpperCase());

// System.out.println("full string: " + Integer.toBinaryString((int) crcResult));

/\*\*

\* The received CRC32 value will have its bits shifted right 24-16-8 times because OutputStream's write(byte)

\* can only write 1 byte at a time, thus the CRC32 will be written to the OutputStream in order

\* and 1 byte at a time.

\*/

for (int j = 24; j >= 0; j-=8){

// System.out.println(j + " : " + Integer.toBinaryString((int)crcResult >> (j)));

os.write((byte)(crcResult >> j));

// System.out.println(Integer.toHexString((int)(crcResult >> j) & 0xFF));

}

int check = is.read();

switch (check){

case 1:

System.out.println("Response good.");

break;

case 0:

System.out.println("Response bad.");

break;

default:

System.out.println("No response");

}

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

}

}

}