/\*\*Travis Vaughn | 4/15/2016

\* File is: JokeServer.java, Version 1.8

\* >javac JokeServer.java

\*

\* In separate shell windows:

\* >java JokeServer

\* >java JokeClient

\* >java JokeClientAdmin

\*

\* Does not matter if client or server is opened first.

\* Server should be started before ClientAdmin is started.

\* ClientAdmin does not need to be run in order to run Client/Server

\*

\* All files needed for running program

\* Travis Vaughn's Joke Server Checklist.html

\* JokeClientAdmin.java

\* JokeServer.java

\* JokeClient.java

\* JokeOutput.docx (to review my results)

\*

\* A server for JokeClient. Elliott, after Hughes, Shoffner, Winslow

\* This will not run unless TCP/IP is loaded on your machine.

\*

\* Notes: For some reason passing an int via BufferedReader causes

\* causes null character. Would get NumberFormatExceptions, but couldn't

\* resolve. When tried parsing string into array and only taking the character

\* for the int, system would throw NullPointerExceptions. Couldn't resolve

\* this either despite allocating new memory for new lists.

\* Finally ended up handling the nullpointer exception by simply ignoring it.

\* Not ideal, but program still works. IF YOU CAN FIGURE OUT THE ISSUE, PLEASE

\* LET ME KNOW AS I WAS LOSING MY MIND ON IT!!! (STARTS @ LINE 109)

---------------------------------------------------------------\*/

import java.io.\*; //Get the Input Output libraries

import java.net.\*; //Get the Java networking libraries

import java.lang.\*; //Get the Java langauge libraries

import java.util.\*; //Get Java utilities libraries

/\*These are the hard-coded jokes and proverbs\*/

class Text{

//all jokes and proverbs are not my own original work.

//they have been taken from the Internet.

//joke variables

//88 is an arbitrary sequence at the end of each mode array in

//JokeServer.java

public static String[] jokeA = new String[]{"A. How many computer programmers does it take to change a light bulb?",

"None, that's a hardware problem.","", "88"};

public static String[] jokeB = new String[]{"B. Listen to this:", "", "Programmer A says: How much money do I owe you?",

"Programmer B says: 500 dollars.",

"Programmer A says: Shall we round it off to 512?", "", "88"};

public static String[] jokeC = new String[]{"C. To understand recursion, one must first understand recursion.", "", "88"};

public static String[] jokeD = new String[]{"D. A feature is nothing more than a bug with seniority.", "", "88"};

public static String[] jokeE = new String[]{"E. Did I ever sing you my new joke song?",

"I've got 99 little bugs in the code,",

"99 little bugs,",

"Take one down, patch it around,",

"I've got 127 little bugs in the code.", "", "88"};

public static String[][] jokeArrays = new String[][]{jokeA, jokeB, jokeC, jokeD, jokeE};

//proverb variables

public static String[] proverbA = new String[]{"A. If you aren't, at any given time, scandalized by code you wrote five" +

" or even three years ago, you're not learning anywhere near enough.","","88"};

public static String[] proverbB = new String[]{"B. The most important property of a program is whether it" +

" accomplishes the intention of its user.", "","88"};

public static String[] proverbC = new String[]{"C. Computer science is no more about computers " +

"than astronomy is about telescopes.", "","88"};

public static String[] proverbD = new String[]{"D. Being an expert can be an excuse for staying in your comfort zone.", "", "88"};

public static String[] proverbE = new String[]{"E. The first symptom of stagnation is preference.", "","88"};

public static String[][] proverbArrays = new String[][]{proverbA, proverbB, proverbC, proverbD, proverbE};

//maintenance variable

public static String[] maintenance = new String[]{"The server is temporarily unavailable--check-back shortly.", "", "88"};

public static String[][] maintenanceArrays = new String[][]{maintenance};

}

/\* This creates a new thread each time it's called \*/

class Worker extends Thread { //Class definition

Socket sock; //Class member, socket, local to Worker

Worker (Socket s) {sock = s;} //Constructor, assign arg s to local sock

/\*This function creates the I/O streams and passes arguments from client to server,

\* and prints back status to both client and server

\*/

public void run(){

//Get I/O streams in/out from the socket:

PrintStream out = null;

BufferedReader in = null;

DataInputStream inInt = null;

try {

//creating the input stream to the server from the client (where it takes in the host joke/ip address

in = new BufferedReader

(new InputStreamReader(sock.getInputStream()));

//creating the output stream where we print out on the server screen what we're looking up

out = new PrintStream(sock.getOutputStream());

// Note that this branch might not execute when expected:

try {

String name, clientID, strIndex;

int serverIndex = 0;

out.println(Mode.currentMode); //sends client current mode

clientID = in.readLine(); //stores client ID

name = in.readLine(); //stores user name

strIndex = in.readLine(); //stores string of server index

try {

serverIndex = Integer.parseInt(strIndex); //converts string index to int index

}catch(NumberFormatException e){} //keeps treating int as NULL; no amount of regex, looping, etc. could solve

printJoke(name, serverIndex, out); //prints details back to client

} catch(IOException x) { //if an IO error when reading in line from client, will catch exception and print below error

System.out.println("Server read error");

x.printStackTrace();

}

} catch(IOException ioe) {} //if error when opening IO streams, will catch exception and print it to screen

}

/\* Prints to client: input joke given, the host joke of the input, and the host IP

\* Also handles if unknown exception occurs

\*/

static void printJoke (String name, int serverIndex, PrintStream out) {

//server output if in joke mode

if (Mode.currentMode.equals("j")){

out.println("Looking up joke...\n");

out.println(name + ",");

for (int i = 0; i < Text.jokeArrays[serverIndex].length; i++) {

out.println(Text.jokeArrays[serverIndex][i]); //To client

}

}

//server output if in proverb mode

else if (Mode.currentMode.equals("p")){

out.println("Looking up proverb...\n");

out.println(name + ",");

for (int j = 0; j < Text.proverbArrays[serverIndex].length; j++) {

out.println(Text.proverbArrays[serverIndex][j]); //To client

}

}

//server out put if in maintenance mode

else if (Mode.currentMode.equals("m")){

for (int k = 0; k < Text.maintenanceArrays[0].length; k++) {

out.println(Text.maintenanceArrays[0][k]); //To client

}

}

//error checker

else {

out.println("We are experiencing technical difficulties right now. The mode is: " + Mode.currentMode);

}

}

}

/\*This runs creates a new admin specific thread\*/

class AdminRequest implements Runnable {

public void run(){ // running admin request

int q\_len = 6; /\* Number of requests for OpSys to queue \*/

int port = 4555; // listening at different port as other port is blocked and should remain block

//just as this one should be.

Socket sock;

try{

ServerSocket servsock = new ServerSocket(port, q\_len);

while (true) {

// listening for admin connection

sock = servsock.accept();

new AdminWorker (sock).start();

}

}catch (IOException ioe) {System.out.println(ioe);}

}

}

/\*This handles the connection with the Admin client by

taking in client request, changing the mode, updating the global

mode and send confirmation back to client

\*/

class AdminWorker extends Thread { //Class definition

Socket sock; //Class member, socket, local to Worker

AdminWorker (Socket s) {sock = s;} //Constructor, assign arg s to local sock

/\*This function creates the I/O streams and passes arguments from client to server,

\* and prints back status to both client and server

\*/

public void run(){

//Get I/O streams in/out from the socket:

PrintStream out = null;

BufferedReader in = null;

try {

//creating the input stream to the server from the admin client

in = new BufferedReader

(new InputStreamReader(sock.getInputStream()));

//creating the output stream where we print out on the server screen what we're looking up

out = new PrintStream(sock.getOutputStream());

// Note that this branch might not execute when expected:

try {

String mode;

mode = in.readLine(); //stores the value input on the client side

System.out.println("Processing mode request: " + mode); //prints out that we're looking up host joke/ ip

changeMode(mode, out); //prints details back to client

} catch(IOException x) { //if an IO error when reading in line from client, will catch exception and print below error

System.out.println("Server read error");

x.printStackTrace();

}

} catch(IOException ioe) {System.out.println(ioe);} //if error when opening IO streams, will catch exception and print it to screen

}

/\* Prints to client: input joke given, the host joke of the input, and the host IP

\* Also handles if unknown exception occurs

\*/

static void changeMode (String mode, PrintStream out) {

out.println("Processing mode request...\n");

if (Mode.currentMode.equals(mode)){

out.println("Server is already in this mode, nothing changed.");

}

else if (mode.equals("j") || mode.equals("p") || mode.equals("m")){

Mode.currentMode = mode;

if (Mode.currentMode.equals("j")) {out.println("Server mode changed to: joke-mode");}

else if(Mode.currentMode.equals("p")) {out.println("Server mode changed to: proverb-mode");}

else if(Mode.currentMode.equals("m")) {out.println("Server mode changed to: maintenance-mode");}

else {out.println("We're experenience difficulties. We appreciate your patience.");}

}

else{

out.println("Incorrect mode entered.");

}

}

}

/\*Global variable to maintain mode to allow for adjustment throughout code\*/

class Mode{

public static String currentMode;

}

public class JokeServer {

/\* This is the code that executes the server

\* Generates infinite loop to keep handling client requests

\* With each new request, a new thread is created to handle request

\*/

public static void main(String a[]) throws IOException{

int q\_len = 6; //Not interesting. Number of requests for Opsys to queue

int port = 4444; //this is how the server and client know where to send/receive requests on the server.

// it specifically identifies the process, since a server can handle multiple different requests

Mode.currentMode = "j";

Socket sock;

AdminRequest admin = new AdminRequest(); // create different thread for admin specific request

Thread t = new Thread(admin);

t.start(); // waits for admin input

ServerSocket servsock = new ServerSocket(port, q\_len); //initializes new server socket

System.out.println

("Travis Vaughn's Joke Server 1.8 starting up.\n");

//this is the main server routine. It listens for client requests and then sends their requests

//off to new threads for the Worker class to handle

while (true){

sock = servsock.accept(); //wait for the next client connection

new Worker(sock).start(); //Spawn worker to handle it; this is the multi-threading

//creating a new thread each time a new client request comes in

}

}

}

/\*\*Travis Vaughn | 4/15/2016

\* File is: JokeClient.java, Version 1.8

\* >javac JokeClient.java

\*

\* In separate shell windows:

\* >java JokeServer

\* >java JokeClient

\* >java JokeClientAdmin

\*

\* If operating across multiple machines, be sure to enter

\* machine's IP address where server is hosted as a command

\* when running JokeClient.

\* >java JokeClient 127.1.0.0

\* Does not matter if client or server is opened first.

\* Server should be started before ClientAdmin is started.

\* ClientAdmin does not need to be run in order to run Client/Server

\*

\* All files needed for running program

\* Travis Vaughn's Joke Server Checklist.html

\* JokeClientAdmin.java

\* JokeServer.java

\* JokeClient.java

\* JokeOutput.docx (to review my results)

\*

\* A client for JokeServer. Elliott, after Hughes, Shoffner, Winslow

\* This will not run unless TCP/IP is loaded on your machine.

\*

---------------------------------------------------------------\*/

import java.io.\*; //Get Input Output libraries

import java.net.\*; //Get the Java networking libraries

import java.util.\*; //Get Java utilities libraries

/\*Global variable for current mode. Used to regularly update mode from server\*/

class Mode{

public static String currentMode = "";

}

/\*Global variable for state. 0 = joke/proverb unseen. 1 = joke/proverb is seen\*/

class State{

public static Integer[] joke = new Integer[]{0,0,0,0,0};

public static Integer[] proverb = new Integer[]{0,0,0,0,0};

}

/\*This is the main joke client\*/

/\*Allows users to connect to server,

\*enter their name, and request jokes/proverbs

\*This is contained code so that multiple clients on multiple

\*machines can interact with the server independently

\*/

public class JokeClient {

/\*Main\*/

public static void main(String args[]) {

String serverName;

UUID clientID = UUID.randomUUID(); //generating unique client ID

//if there are no arguments after program name, assume it's localhost

//otherwise set server to IP/name given

if(args.length < 1) serverName = "localhost";

else serverName = args[0];

System.out.println("Travis Vaughn's Joke Client, 1.8.\n");

System.out.println("Using server: " + serverName + ", Port: 4444\n"); //not worth it to soft-code.

//according to Java documentation, this is a wrapper around Reader which allows

//for most efficient handling of reading lines from inputs

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

//this is the main client interaction

try {

String request, username;

int requestNum = 0;

//getting user name

System.out.print("What is your name? ");

username = in.readLine(); //gets user name

System.out.print("Hello " + username + "! Ready for some fun?!\n");

do {

//asking user to request a joke

System.out.print

("Press Enter for a joke or proverb :), (\\quit) to end: "); //included ability to quit

System.out.flush();

request = in.readLine(); //storing user entry into variable

getMode(serverName); //getting mode from server

//this is how we allow user to quit

//i included '\' in front of quit as I believe it's not a valid URL char

//even if it is, users are at least less likely to enter \quit for a website

//as they are quit

if (request.indexOf("\\quit") < 0) {

int serverIndex = getIndex();

getJoke(username, serverIndex, serverName, clientID.toString()); //this sends the website we wish to visit to the server

}

} while (request.indexOf("\\quit") < 0); //exits client server if user enters "\quit"

System.out.println("Cancelled by user request.");

} catch(IOException x) {x.printStackTrace();} //throws IO exception if issue with print out or, storing names, or getting remote address

}

/\* sends a hostname/ip to a server where the server then returns back results to client

\* See server side code for what it sends back to client

\*/

static void getJoke(String name, int serverIndex, String serverName, String clientID){

Socket sock;

BufferedReader fromServer;

PrintStream toServer;

String textFromServer;

try{

//Open our connection to server port, choose your own port number..

sock = new Socket(serverName, 4444); //this is client setting up ability to communicate with server

//Create filter I/O streams for the socket:

fromServer =

//opening input stream from server; sock has our server name and port number

new BufferedReader(new InputStreamReader(sock.getInputStream()));

toServer = new PrintStream(sock.getOutputStream()); //opening output stream to our server

//sock has our server name and port number

//Send request to server:

toServer.println(clientID);

toServer.println(name);

toServer.println(serverIndex);

toServer.flush();

fromServer.readLine(); //taking mode off printline

//continue reading and printing text from server until you find the

//special 88 escape characters

while(!(textFromServer = fromServer.readLine()).equals("88")){

System.out.println(textFromServer);

}

sock.close(); //close connection with server

} catch (IOException x){ //throws IO exception if IO error occurs either when open i/o streams, closing streams, or passing requests

System.out.println("Socket error.");

x.printStackTrace();

}

}

static void getMode(String serverName){

Socket sock;

BufferedReader fromServer;

try{

sock = new Socket(serverName, 4444); //this is client setting up ability to communicate with server

//Create filter I/O streams for the socket:

fromServer =

//opening input stream from server; sock has our server name and port number

new BufferedReader(new InputStreamReader(sock.getInputStream()));

Mode.currentMode = fromServer.readLine();

sock.close(); //close connection with server

} catch (IOException x){ //throws IO exception if IO error occurs either when open i/o streams, closing streams, or passing requests

System.out.println("Socket error.");

x.printStackTrace();

}

}

/\*Looks at the mode and the state and returns a index to send

to the server in order to retrieve joke/proverb

\*/

static int getIndex(){

//if in joke mode

if (Mode.currentMode.equals("j")){

int jokeIndex, rand;

//breaks only when we have an unseen index

while(true){

ArrayList<Integer> unseenJokeIndeces = new ArrayList<Integer>();

//create array of unseen jokes

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

if(State.joke[i] == 0){

unseenJokeIndeces.add(i); //add unseen joke

}

}

if (unseenJokeIndeces.size() > 0){ //if there is at least one unseen joke

//generate random index from unseenjokes

rand = (int)(Math.random() \* (unseenJokeIndeces.size()) + 0);

System.out.println("This is rand: " + rand);

jokeIndex = unseenJokeIndeces.get(rand); //stores joke index

break; //end loop

}

else{ //if all jokes have been seen

for (int ii = 0; ii < State.joke.length; ii++){

State.joke[ii] = 0; //reset all values

}

}

}

State.joke[jokeIndex] = 1; //set joke to seen

return jokeIndex;

}

//if in proverb mode

else if (Mode.currentMode.equals("p")){

int proverbIndex, rand;

//breaks only when we have an unseen index

while(true){

ArrayList<Integer> unseenProverbIndeces = new ArrayList<Integer>();

//create array of unseen jokes

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

if(State.proverb[i] == 0){

unseenProverbIndeces.add(i); //add unseen joke

}

}

if (unseenProverbIndeces.size() > 0){ //if there is at least one unseen joke

//generate random index from unseenjokes

rand = (int)(Math.random() \* (unseenProverbIndeces.size()-1) + 0);

proverbIndex = unseenProverbIndeces.get(rand); //stores joke index

break; //end loop

}

else{ //if all jokes have been seen

for (int ii = 0; ii < State.proverb.length; ii++){

State.proverb[ii] = 0; //reset all values

}

}

}

State.proverb[proverbIndex] = 1; //set joke to seen

return proverbIndex;

}

else {return 0;}

}

}

/\*\*Travis Vaughn | 4/15/2016

\* File is: JokeClientAdmin.java, Version 1.8

\* >javac JokeClientAdmin.java

\*

\* In separate shell windows:

\* >java JokeServer

\* >java JokeClient

\* >java JokeClientAdmin

\*

\* Ensure that JokeServer is running before

\* submitting request; otherwise no changes will be made.

\*

\* All files needed for running program

\* Travis Vaughn's Joke Server Checklist.html

\* JokeClientAdmin.java

\* JokeServer.java

\* JokeClient.java

\* JokeOutput.docx (to review my results)

\*

\* A client for JokeServer. Elliott, after Hughes, Shoffner, Winslow

\* This will not run unless TCP/IP is loaded on your machine.

\*

---------------------------------------------------------------\*/

import java.io.\*; //Get Input Output libraries

import java.net.\*; //Get the Java networking libraries

/\*Allows client to change mode to joke, proverb, or maintenance

\*Server makes changes and this impacts all clients connected

\*/

public class JokeClientAdmin {

public static void main(String args[]) {

String serverName;

//if there are no arguments after program name, assume it's localhost

//otherwise set server to IP/name given

if(args.length < 1) {serverName = "localhost"; }

else {serverName = args[0];}

System.out.println("Travis Vaughn's Joke Client Admin, 1.8.\n");

System.out.println("Using server: " + serverName + ", Port: 4555\n"); //not worth it to soft-code.

//according to Java documentation, this is a wrapper around Reader which allows

//for most efficient handling of reading lines from inputs

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

try {

String mode;

System.out.print("Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' "+

"for maintenance mode: ");

mode = in.readLine(); //gets user name

sendMode(mode, serverName); //this sends the mode we wish to change the server to

}catch(IOException x) {x.printStackTrace();} //throws IO exception if issue with print out or, storing names, or getting remote address

}

/\* sends a hostname/ip to a server where the server then returns back results to client

\* See server side code for what it sends back to client

\*/

static void sendMode(String mode, String serverName){

Socket sock;

BufferedReader fromServer;

PrintStream toServer;

String textFromServer;

try{

//Open our connection to server port, choose your own port number..

sock = new Socket(serverName, 4555); //this is client setting up ability to communicate with server

//Create filter I/O streams for the socket:

fromServer =

//opening input stream from server; sock has our server name and port number

new BufferedReader(new InputStreamReader(sock.getInputStream()));

toServer = new PrintStream(sock.getOutputStream()); //opening output stream to our server

//sock has our server name and port number

//Send request to server:

toServer.println(mode);

toServer.flush();

//Read two or three lines of response from the server,

//and block while synchronously waiting;

for (int i = 1; i <=3; i++){

textFromServer = fromServer.readLine();

if(textFromServer != null) System.out.println(textFromServer);

}

}catch (IOException ioe) {System.out.println(ioe);}

}

}

<html>

<head> <title> Travis Vaughn's Joke Server Checklist </title>

</head>

<body>

<FONT FACE="Cambria" SIZE=1>

2.1

<br>

Updates:

<ol>

<li> none

</ol>

<FONT SIZE=3>

<!-- Copy this template into /joke/checklist.html, add name, change no to -->

<!-- yes when you complete that task. Read the class contract on checklists!-->

<h1>

<center> Program One Checklist -- Multithreaded Joke Server and Client </center> <p>

</h1>

<FONT FACE="Cambria" SIZE=4>

Required, <i> <strong> precisely named </strong> </i> submission files, NO

subdirectories, NO Packages:

<ul>

<li> InetClient.java

<li> InetServer.java

<li> JokeClient.java

<li> JokeServer.java

<li> JokeClientAdmin.java

<li> JokeOutput.txt <-- concatenate all output with comments.

</ul>

Change "No" to "Yes" as you complete the tasks:<p>

<table border=1 width="80%" >

<tr><td align="Left"><b> <FONT FACE="Cambria" SIZE=5>Task

</font></b></td>

<td align="Left"><b><FONT FACE="Cambria" SIZE=5> Complete? </task>

</b></td></tr>

<tr> <td align="left"><b>

<center> ADMINISTRATION </center> </b></td><td align="left"><b>

</b></td></tr>

<tr> <td align="left"><b>

My name is... </b></td><td align="left"><b>

Travis Vaughn </b></td></tr>

<tr> <td align="left"><b> Program Files submitted to D2L as <i> standard ZIP

file </i> only

</b></td><td align="left"><b> Yes </b></td></tr>

<tr> <td align="left"><b> All JokeServer code (no Inet code!) concatenated

with output and submitted to D2L TII link as .docx or .html only</b></td><td align="left"><b> Yes

</b></td></tr>

<tr> <td align="left"><b>

All files named correctly (see above) </b></td><td align="left"><b>

Maybe </b></td></tr>

<tr> <td align="left"><b>

All files in ONE directory (no subdirs!) </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Included the correct headers for source files </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Class contract submitted </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

<center> PROGRAMS </center> </b></td><td align="left"><b>

</b></td></tr>

<tr> <td align="left"><b>

Basic Inet Client and Server Run </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Extensive Comments Inet </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Extensive Comments JokeServer </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Tested with this many clients simultaneously (more than one!) </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Returns five jokes </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Returns jokes randomly </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Labels the jokes and proverbs A-E </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Returns five proverbs </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Returns proverbs randomly </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Performs name substitution </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Admin Client connects at different port </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Switches server mode from Joke to Proverb to Maintenance </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Maintains Proverb state, and does not repeat </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Maintains Joke state, and does not repeat </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Maintains converstations with interleaved Joke/Proverb modes </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Only have to type user name once </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

<center> BRAGGING RIGHTS BELOW </center> </b></td><td align="left"><b>

THIS LINE </b></td></tr>

<tr> <td align="left"><b>

Tested Running between different machines </b></td><td align="left"><b>

Yes </b></td></tr>

<tr> <td align="left"><b>

Writes and restores state from disk </b></td><td align="left"><b>

No </b></td></tr>

<tr> <td align="left"><b>

Wrote thread-safe code </b></td><td align="left"><b>

No </b></td></tr>

<tr> <td align="left"><b>

Uses a secure channel for administration </b></td><td align="left"><b>

No </b></td></tr>

<tr> <td align="left"><b>

[Fill in custom extra features, one line per feature] </b></td><td align="left"><b>

Yes </b></td></tr>

</table>

<h2> Optional Comments: </h2>

[Communicate with us about this program here if necessary.]<br><br>

Explain <i> Maybe </i> entries here also.

Uncertain what concatenated file should be named as didn't see it specified in instructions.

</body></html>

**\*\*\*\*\*RUNNING OUTPUT PER INSTRUCTIONS\*\*\*\*\***

**1. /\*THIS IS THE FIRST RUN ON ONLY ONE CLIENT\*/**

**\*\*\*CLIENT\*\*\***

C:\Users\TravTron\Desktop\School>javac JokeClient.java

C:\Users\TravTron\Desktop\School>java JokeClient

Travis Vaughn's Joke Client, 1.8.

Using server: localhost, Port: 4444

What is your name? Travis

Hello Travis! Ready for some fun?!

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

E. Did I ever sing you my new joke song? Never repeats {B, A, D, C, E}

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

E. Did I ever sing you my new joke song? I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

A. How many computer programmers does it take to change a light bulb? Never repeats but diff. order {C, B, E, D, A}

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

E. The first symptom of stagnation is preference. Never repeats {A, B, C, D, E}

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

E. The first symptom of stagnation is preference.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

D. Being an expert can be an excuse for staying in your comfort zone. Never repeats but diff. order {C, B, E, A, D}

Press Enter for a joke or proverb :), (\quit) to end:

**\*\*\*CLIENT ADMIN\*\*\***

C:\Users\TravTron\Desktop\School>javac JokeClientAdmin.java

C:\Users\TravTron\Desktop\School>java JokeClientAdmin

Travis Vaughn's Joke Client Admin, 1.8.

Using server: localhost, Port: 4555

Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' for maintenance mode: p

Processing mode request...

Server mode changed to: proverb-mode

C:\Users\TravTron\Desktop\School>

**\*\*\*SERVER\*\*\***

C:\Users\TravTron\Desktop\School>javac JokeServer.java

C:\Users\TravTron\Desktop\School>java JokeServer

Travis Vaughn's Joke Server 1.8 starting up.

Processing mode request: p

**2. /\*THIS IS THE SECOND RUN WITH TWO CLIENTS\*/**

**\*\*\*CLIENT 1\*\*\***

C:\Users\TravTron\Desktop\School>java JokeClient

Travis Vaughn's Joke Client, 1.8.

Using server: localhost, Port: 4444

What is your name? Travis

Hello Travis! Ready for some fun?!

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

E. Did I ever sing you my new joke song?

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

E. Did I ever sing you my new joke song?

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

E. The first symptom of stagnation is preference.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Travis,

E. The first symptom of stagnation is preference.

Press Enter for a joke or proverb :), (\quit) to end:

**\*\*\*CLIENT 2\*\*\***

C:\Users\TravTron\Desktop\School>java JokeClient

Travis Vaughn's Joke Client, 1.8.

Using server: localhost, Port: 4444

What is your name? Justin

Hello Justin! Ready for some fun?!

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

E. Did I ever sing you my new joke song?

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

B. Listen to this: **I interleaved one after the other on client 1 vs. client 2; so they both got B on the third try and different items on the other tries.**

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

E. Did I ever sing you my new joke song?

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

E. The first symptom of stagnation is preference.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up proverb...

Justin,

E. The first symptom of stagnation is preference.

Press Enter for a joke or proverb :), (\quit) to end:

**\*\*\*CLIENT ADMIN\*\*\***

C:\Users\TravTron\Desktop\School>java JokeClientAdmin

Travis Vaughn's Joke Client Admin, 1.8.

Using server: localhost, Port: 4555

Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' for maintenance mode: p

Processing mode request...

Server mode changed to: proverb-mode

C:\Users\TravTron\Desktop\School>

**\*\*\*SERVER\*\*\***

C:\Users\TravTron\Desktop\School>java JokeServer

Travis Vaughn's Joke Server 1.8 starting up.

Processing mode request: p

**3. /\*THIS IS THE FINAL RUN WITH TWO CLIENTS INTERLEAVING PROVERB MODE AND JOKE MODE\*/**

**You’ll notice throughout this last one that each client switched to each mode correctly and switched back. Additionally each client had their own separate random lists they were using.**

**\*\*\*CLIENT 1\*\*\***

C:\Users\TravTron\Desktop\School>javac JokeClient.java

C:\Users\TravTron\Desktop\School>java JokeClient

Travis Vaughn's Joke Client, 1.8.

Using server: localhost, Port: 4444

What is your name? Travis

Hello Travis! Ready for some fun?!

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Travis,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Travis,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Travis,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Travis,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Travis,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Travis,

E. Did I ever sing you my new joke song?

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Travis,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Travis,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

The server is temporarily unavailable--check-back shortly.

Press Enter for a joke or proverb :), (\quit) to end:

The server is temporarily unavailable--check-back shortly.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Travis,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Travis,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

**\*\*\*CLIENT 2\*\*\***

C:\Users\TravTron\Desktop\School>java JokeClient

Travis Vaughn's Joke Client, 1.8.

Using server: localhost, Port: 4444

What is your name? Justin

Hello Justin! Ready for some fun?!

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Justin,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Justin,

A. How many computer programmers does it take to change a light bulb?

None, that's a hardware problem.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Justin,

D. Being an expert can be an excuse for staying in your comfort zone.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Justin,

C. Computer science is no more about computers than astronomy is about telescope

s.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Justin,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Justin,

C. To understand recursion, one must first understand recursion.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up joke...

Justin,

E. Did I ever sing you my new joke song?

I've got 99 little bugs in the code,

99 little bugs,

Take one down, patch it around,

I've got 127 little bugs in the code.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Justin,

B. Listen to this:

Programmer A says: How much money do I owe you?

Programmer B says: 500 dollars.

Programmer A says: Shall we round it off to 512?

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **joke...**

Justin,

D. A feature is nothing more than a bug with seniority.

Press Enter for a joke or proverb :), (\quit) to end:

The server is temporarily unavailable--check-back shortly.

Press Enter for a joke or proverb :), (\quit) to end:

The server is temporarily unavailable--check-back shortly.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Justin,

B. The most important property of a program is whether it accomplishes the inten

tion of its user.

Press Enter for a joke or proverb :), (\quit) to end:

Looking up **proverb...**

Justin,

A. If you aren't, at any given time, scandalized by code you wrote five or even

three years ago, you're not learning anywhere near enough.

Press Enter for a joke or proverb :), (\quit) to end:

**\*\*\*CLIENT ADMIN\*\*\***

C:\Users\TravTron\Desktop\School>java JokeClientAdmin

Travis Vaughn's Joke Client Admin, 1.8.

Using server: localhost, Port: 4555

Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' for maintenance mode: p

Processing mode request...

Server mode changed to: proverb-mode

C:\Users\TravTron\Desktop\School>java JokeClientAdmin

Travis Vaughn's Joke Client Admin, 1.8.

Using server: localhost, Port: 4555

Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' for maintenance mode: j

Processing mode request...

Server mode changed to: joke-mode

C:\Users\TravTron\Desktop\School>java JokeClientAdmin

Travis Vaughn's Joke Client Admin, 1.8.

Using server: localhost, Port: 4555

Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' for maintenance mode: m

Processing mode request...

Server mode changed to: maintenance-mode

C:\Users\TravTron\Desktop\School>java JokeClientAdmin

Travis Vaughn's Joke Client Admin, 1.8.

Using server: localhost, Port: 4555

Enter 'j' for joke-mode, 'p' for proverb-mode or 'm' for maintenance mode: p

Processing mode request...

Server mode changed to: proverb-mode

C:\Users\TravTron\Desktop\School>

**\*\*\*SERVER\*\*\***

C:\Users\TravTron\Desktop\School>javac JokeServer.java

C:\Users\TravTron\Desktop\School>java JokeServer

Travis Vaughn's Joke Server 1.8 starting up.

Processing mode request: p

Processing mode request: j

Processing mode request: m

Processing mode request: p