**CNT 4007 Computer Network Fundamentals**

**Project 3 Individual Project**

**Internet Chatting**

1. **Description**

Write a peer-to-peer chat program, which will create two threads: a main thread and a new thread created by the main thread with new Handler(…).start(); refer to the server code of project 2 for sample code. The main thread is used as *the writing thread* where you write chat messages to your friend. The created thread is used as the *reading thread* where you receive chat messages from your friend.

You will submit a single program, chat, consisting of a server part, i.e., the reading thread, and a client part, i.e., the writing thread. You need to run this program twice in two windows respectively, one referred to as Alice and the other as Bob:

In Alice window: chat port1

In Bob window: chat port2

So, chat takes one argument, the port number where the reading thread will be listening to (and reading messages from). To pass the script, you should use

In Alice window: chat 5106

In Bob window: chat 4007

such that Alice listens to 5106, to which Bob will connect later, and Bob listens to 4007, to which Alice will connect later, for two-way communication.

**Execution of Reading Thread:** It creates a ServerSocket with the port number from the argument and then listens on the socket for new connection, using the accept method. When a new connection from the writing thread of another user arrives, it will read messages from the connection socket that is created --- the code resembles the server code of project 1, not project 2. It prints out all received messages.

**Execution of Writing Thread:** It first reads from keyboard for the port number that the reading thread of the other peer is listening to, and then creates a socket connecting to that port. After the connection (socket) is successfully established, it goes into a loop of reading a message from the keyboard and writing the message to the connection (socket).

We will test your program using a script with the following steps:

1. run “chat 5106” in a window for Alice
2. run “chat 4007” in another window for Bob
3. Alice types “4007”
4. Bob types “5106”
5. Alice types “Hello Bob xxxx” where xxxx is a four digit random number. The message should show up in Bob’s window.
6. Bob types “Hello Alice yyyy” where yyyy is another four digit random number. The message should show up in Alice’s window.

To pass the script, please follow exactly the above steps, with the commands/ports/messages in quote. Try not to include extra or different input/output, so that you have a better chance to pass the script. Java and Python are highly preferred because they are platform-independent.

1. **Summary of Details**

**Implementation:**

**chat:**

* + Take one argument x as port number
  + Two threads, one listening on a server-side port (which is x), the other connecting to another user (which is supplied from keyboard by user)
  + Chat with another user

**Compilation:**

**For Java:**

javac chat.java

**For Python:** (No compilation needed)

**For C++:**

g++ chat.cpp -o chat

**Execution for Self-Testing:**

1. **Start Alice and Bob:**

* **For Java:**
  + - java chat 5106
    - java chat 4007
* **For Python:**
  + - python chat.py 5106
    - python chat.py 4007
* **For C++:**
  + - ./chat 5106
    - ./chat 4007

1. **Establishing connections**
   * In Alice’s window:
   * Type “4007”
   * In Bob’s window:
   * Type “5106”
2. **Chat Testing:**
   * In Alice’s window:
   * Type “Hello Bob xxxx” where xxxx is a four digit random number
   * In Bob’s window:
   * Type “Hello Alice yyyy” where yyyy is a four digit random number

**Files:**

* Upon submission, all project files must be packaged into a .zip, .rar, or .tar.gz archive.
* The archive should either contain no directories or a single top-level directory.

By following these guidelines precisely, your project will be compatible with the script.

1. **Programming Environment**

Programming language: Java, C, C++, C#, Python

Operating System: Windows, Mac OS or Linux

Programming Tool: Eclipse, IntelliJ, Jcreator, Kawa, Netbeans, … whatever you like.

To use Eclipse, please go through the following list:

1. Download JDK from: <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

2. Download Eclipse from: <http://www.eclipse.org/downloads/>

3. Here is a link for eclipse tutorial: <http://eclipsetutorial.sourceforge.net/totalbeginner.html>

4. Here is a tutorial for socket programming in Java: <http://java.sun.com/docs/books/tutorial/networking/sockets/>

1. **Code Submission**

If you use Java, you will need to submit the following files: chat.java, chat.class, in a zipped directory, e.g., project3.rar or project3.zip.

If you use C/C++ or Python please put all source files and executables in a zipped directory.

If you are an online student, you must include README.txt to explain how a TA will run your program, particularly if you use C/C++.

Submit the project through Canvas:

1. Go to <https://lss.at.ufl.edu/>
2. Click “Login to e-Learning”
3. Login with your gator link username/password
4. Go in CNT 4007
5. Click “Assignments” and submit your project

This is an individual project. Students must submit their code via Canvas by the deadline. We will run an automatic tool to catch submissions with identical or similar code. There will be no late submissions allowed.