UNIVERSITY *of*
NEW ORLEANS

DEPARTMENT OF COMPUTER SCIENCE

CSCI 4311/5311 Computer Networks and Telecommunications

Fall 2017

Programming Assignment #1 (PA1): `netcat`

Assigned: Thu, Sep 14, 2017

Due: Sunday, October 1, 2017@11:59pm

Objective:

This is an introductory project, which builds on the example socket API code shown in class. The end product of the assignment will be a simple *Java* implementation of the popular `netcat` (`nc`) tool (http://netcat.sourceforge.net/), which provides a lightweight mechanism to transfer data across the network.

User requirements:

You must build two pieces of code—a netcat client/server pair. The two have almost identical functionality and only differ in the way they are invoked.

- ***Netcat server*** binds to a specified port number on the local host and waits for a connection request from a client. Once a connection is established, it operates in one of two modes—download and upload.
    - In download mode, it reads data from its standard input device and writes it to the socket.
    - In upload mode, it reads data from the socket and writes it to standard output.
- ***Netcat client*** establishes a connection to the server on the given host name (or IP address) and port number and operates in one of two modes, opposite the server:
    - In download mode, it reads data from the socket and writes it to standard output.
    - In upload mode, it reads data from its standard input device and writes it to the socket.

Command line usage:

Download a file from hostA to hostB:

```
hostA> java csci4311.nc.NetcatServer 1234 < original-file
hostB> java csci4311.nc.NetcatClient hostA 1234 > downloaded-file
```

Upload a file from hostB to hostA:

```
hostA> java csci4311.nc.NetcatServer 4321 > uploaded-file
hostB> java csci4311.nc.NetcatClient hostA 4321 < original-file
```

Implementation requirements

- This is an individual assignment—the submitted code must yours, and yours only. (*You can use snippets of example code discussed in class.*)
- You must have two main classes `csci4311.nc.NetcatServer` and `csci4311.nc.NetcatClient` that can be executed as specified above.
- You can have as many other classes as you need but ***all*** must be members of the `csci4311.nc` package.
- You are only allowed to use classes that are part of the standard JDK (`java.*` packages).
- Hint: you can use the `System.in.available()` method to test if standard input has been redirected—if yes, result should be positive, otherwise zero.

Submission

- ***No late submissions***. You have two weeks, which is significantly more time than you need, so that you can plan your schedule and accommodate various commitments.
- Name your submission files as required above (along with any other files you need). ***Only submit source code.***
- Create a **private** repo named `<your-login>/csci4311-f17` on `gitlab.cs.uno.edu`. **If this repo is not private I will consider this cheating (sharing answers) and you will receive no credit for this assignment!**
- `git push` it to your private repo on `gitlab.cs.uno.edu:<your-login>/csci4311-f17` and make sure that the instructor (`jtsylve`) must have *reporter* access.

  Note: Make sure your repo is named ***exactly*** as specified in the convention above. At the end of the semester, the repos will be garbage collected (you will be given notice to archive your work).
- Place your submission in a `/pa1` folder.
- Your commit message must read "`submission: pa1`".
- **If you do not follow these submission instructions exactly, your submission will not be graded and you will receive a 0.**

Scoring:

This PA is worth 50 points.

Extra Credit (*mandatory* for 5311 students)

- **+20%**: Create a UDP implementation of *netcat*. Specifically, create `csci4311.nc.NetcatUDPServer` and `csci4311.nc.NetcatUDPClient` that function the same way but use UDP as the underlying communication mechanism.

*You should only attempt extra credit **after** you have successfully completed the primary assignment.*