NETS 150 – Homework 3

Due – Apr 1, 2014 at 12.00pm

Part 1 – Theory (20 points)

Please do the following problems:

Paper Programs – Exercise 3 below requires a paper program. This program does not need to be compiled, executed, debugged, etc. Computer generated code will earn no extra credit and will have no effect on your grade.

1. What is a socket? What is the difference between a `Socket` object and a `ServerSocket` object? (5 points)
2. What happens if the `Socket` constructor's second parameter is not the same as the port number at which the server waits for connections? (5 points)
3. Paper Program – modify the Client Server code from class (posted in canvas) to that it supports basic Math. The client should be able to send the server the following commands: ADD X Y, MUL X Y. X and Y can be any 2 doubles. The server should do the math and return the answer to the client. (10 points)

Part 2 – Programming (80 points)

Getting and Analyzing Data from the Internet – The CIA World Factbook

The CIA has an excellent collection of detailed information about each country in the world. It's called the CIA World Factbook. You can find it here https://www.cia.gov/library/publications/the-world-factbook/

For this assignment, you'll write a program in Java to get and analyze data from the CIA World Factbook website. You should not download any of the information to a local file and read from it. Your program must interact directly with the website. (This way your programs will work even when the CIA updates the Factbook, which is done every year.)

Once you have a basic program that can interact with CIA World Factbook, use it to answer the following questions. Describe in detail the algorithm you used and the answers in your readme.txt file.

1. List countries in *South America* that are prone to *earthquakes*.
2. Find the country with the lowest elevation point in *Europe*.
3. List all countries in the *southeastern* hemisphere.
4. List countries in *Asia* with more than *10* political parties.
5. Find the top *5* countries with the highest electricity consumption per capita. (Electricity consumption / population)

6. Certain countries have one dominant religion (in terms of fraction of the population) whereas other countries don't. List countries (along with the religion) where the dominant religion accounts for more than *80%* of the population. List countries (along with the religions) where the dominant religion accounts for less than *50%* of the population.
7. A landlocked country is one that is entirely enclosed by land. For example, Austria is landlocked and shares its borders with Germany, Czech Republic, Hungary, etc. There are certain countries that are entirely landlocked by a single country. Find these countries.
8. Wild card – come up with an interesting question. List the question and find the answer to it.

Note: For the italicized parts in the above, your code should be able to deal with any similar input (e.g., from a user). This should not be hard coded.

Hint: It might be easier to use the text/low bandwidth version of the website in your program.

Part 2 – Extra Credit (10 points)

In addition to the questions above, answer the following questions:

1. I want to go on a vacation with a friend. Our goal is to visit as many capital cities as we can in as short a geographical distance as possible. To make things easier (and not worry about spherical geometry), we are fine with travelling to capitals that are within *10* degrees of latitude and longitude of each other. Find the lat/long coordinates and the list of countries/capitals so that the number of capitals is maximized.
2. Wild card – come up with an interesting question. List the question and find the answer to it.

For the EC part, you cannot have any help from the TAs/instructor.

Grading Criteria (for the programming part)

10% for compilation – If your code compiles, you get full credit. If not, you get a 0.
70% for functionality – Does the code work as required? Does it crash while running? Are there bugs? …
10% for design – Is your code well designed? Does it handle errors well? Do you have a lot of URLs hardcoded? To keep your code flexible, you should hardcode as few URLs as possible in the code.
10% for style – Do you have good comments in the code? Are your variables named appropriately? …

Programming – General Comments

Here are some guidelines wrt programming style for full credit.

Please use Javadoc-style comments.

For things like naming conventions, please see
http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html

You can also install the Checkstyle plugin (http://eclipse-cs.sourceforge.net/) in Eclipse, which will automatically warn you about style violations.

Submission Instructions

We recommend submitting the theory part electronically also. However, you can turn in a physical copy at the start of class, if you prefer. The code should be submitted electronically. Please **do not** print it out.

In addition to the Java files (and the theory writeup), you should also submit a text file titled readme.txt, which should contain a short write-up about your software. How to run your program, any problems you experienced, etc. Think of the readme as a combination of instructions for the user and a chance for you to get partial credit.

Please create a folder called YOUR_PENNKEY. Places all your files inside this – the java files, theory writeup, the readme.txt file. Zip up this folder. It will thus be called YOUR_PENNKEY.zip. So, e.g., my homework submission would be swapneel.zip. Please submit this zip file via canvas.