Advanced Programming, Spring 2016
Homework Exercise 5

Due date: 2/6/2016

General instructions: Fork the repository at `https://ap.idc.ac.il/rhodecode/hw5/first` to a new repository named username-hw5 in the hw5 group. **Make sure to remove read permissions for everyone else after you clone.** This repository contains three classes that you will fill out during the exercise.

1. The repository contains a skeleton for the class `net.HelloServer`. Implement the methods that have a "TODO" comment in them to match their javadoc documentation.
2. The repository contains a skeleton for the class `net.HelloClient`. Implement the methods that have a "TODO" comment in them to match their javadoc documentation.
3. The repository contains a skeleton for the class `net.FileServer`. Implement the methods that have a "TODO" comment in them to match their javadoc documentation.
4. (15 points) **Bonus Question (hard)**: The bonus is implementing a DNS resolver. Implement the methods that have a "TODO" comment in the `net.DNSResolver` class. This will require you to read and understand:
   - The Java documentation for using UDP sockets. This tutorial can help: `https://docs.oracle.com/javase/tutorial/networking/datagrams/index.html`, and the JavaDoc for `DatagramSocket` and `DatagramPacket` will be useful: `https://docs.oracle.com/javase/7/docs/api/java/net/DatagramSocket.html`, `https://docs.oracle.com/javase/7/docs/api/java/net/DatagramPacket.html`).
   - The DNS query and response formats. The official description is RFC 1035 (`https://tools.ietf.org/html/rfc1035`), however, you can find other descriptions on the internet too (this one is not bad: `https://technet.microsoft.com/en-us/library/dd197470%28v=ws.10%29.aspx`). To help with one of the more complex parts of the specification, we have included the `readName` method that can read "compressed" domain names (as they appear in the DNS response—see §4.1.4 of the RFC for info).

Notes:

1. To help you check your code, some of the classes have implementations of the `main` method. You can run these classes and check with ncat or a browser (for the `FileServer` class) if your code works. For debugging the DNS resolver, wireshark can be a big help.
2. Commit early, commit often, and push your changes. You don't have to finish solving a question completely to commit your work. In fact, it is a good idea to commit often, and to push your changes to the rhodecode repository. This will help you if files get accidentally deleted, your computer catches fire, or your goat eats your homework.
3. Remember to comment your code. Part of your grade will be based on code clarity and good comments. This is especially critical if your code has errors (in this case, good comments that show you understood what needs to be done will help you).