**Lab**​​**10**​​**-**​​**Networking**​​**-**​​**Client/Server**

This ​ lab​ ​ contains​ ​ in-class​ ​ exercises​ ​ ​related ​ to​​ networking​ ​ in​ ​ Java.​ ​ Before​ ​ starting​ ​ ​solving these ​ exercises,​ ​ one​ ​ is​ ​ advised​ ​ to​​ review​ ​**j**ava.net.ServerSocket​ ​ and​​ java.net.Socket classes.

**Task** ​ **1:**​​Develop ​ a​ ​ class​ ​ to​ ​ determine​ ​all​ ​ the​ ​ IP​ ​ addresses​ ​ of​ ​ a​ ​ host,​ ​ given​ ​ the​ ​ host's​​ name.​

**Task** ​ **2:**​ ​ Develop​ ​ a​ ​ ​program ​ ​called ​ReadSites​​ that​ takes​ ​ as​ ​ command​ ​ line​​ parameters​ the ​ URL​ ​ strings​ ​ (​ without ​ protocol,​ ​ i.e.​ ​ http://).​ ​ The​​ program​ ​ must​ ​ print​ ​ the​ ​ content​ ​ of​ ​ all​ sites ​ entry​ ​ point​ ​ (​ index.html ​ or​ ​ similar​ ​ for​ ​ each​​ site).​

**Task** ​ **3:**​ ​ Develop​ ​ a​ ​ “Echo”​ ​ multi-threaded​ ​ server.​ ​ The​ ​ server​ ​ structure​​ is​ ​ given​ ​ here:​

public​ ​class​ ​ThreadedEchoServer​ ​{

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

​ ​ServerSocket​ ​s​ ​=​ ​null;

​ ​try​ ​{

​ ​s​ ​=​ ​new​ ​ServerSocket(7777);

​ ​int​ ​i​ ​=​ ​0;

​ ​for​ ​(;​ ​;​ ​)​ ​{

​ ​Socket​ ​incoming​ ​=​ ​s.accept();

​ ​System.out.println("Spawning​ ​"​ ​+​ ​i++);

​ ​new​ ​ThreadedEchoHandler(incoming,​ ​i).start();

​ ​}

​ ​}​ ​catch​ ​(Exception​ ​e)​ ​{

​ ​System.out.println(e);

​ ​}​ ​finally​ ​{

​ ​s.close();

​ ​}

​ ​}

**}**

Study ​ first​ ​ the​ ​Echo ​ server.​​ You​ ​ need​ ​ to​ ​ implement​ ​ the​​ thread​ ​ class​ ​ThreadedEchoHandler

@​ ​Jordan​ ​Anastasiade

**Task** ​ **4:**​ ​ ​ In ​ the​ ​ observer​ ​ example​ ​ provided,​ ​ replace​ ​the​ ​OnlineLearner​and InclassLearner ​ class​ ​ with​ ​ the​ ​ generic​ ​ class​

public​ class​ **Learner<L>**​ ​{

​ ​... }

What else needs to be changed? Refactor the given example so that you can create three learner ​ types.​

@​ ​Jordan​ ​Anastasiade