5 • Networking with Sockets

public HttpConnectionMgr(Socket clientSocket)
{
    fClientSocket = clientSocket;
    this.start();
} //HttpConnectionMgr

/**< *
 * This method executes the core connection-mgmt stuff *
 */
public void run()
{
    try {
        if (fDebugOn) System.out.println("building iostreams...");
        fClientInputStream =
            new DataInputStream(fClientSocket.getInputStream());
        fClientOutputStream =
            new DataOutputStream(fClientSocket.getOutputStream());

        if ((fClientOutputStream != null) && (fClientInputStream != null)) {
            // now, handle the transaction
            fTransactionHandler =
                new HttpTransactionHandler(fClientInputStream, fClientOutputStream);
            try {
                fTransactionHandler.handleTransaction();
            } catch (Exception handleTransEx) {
                System.err.println("handleTransaction ex: " + handleTransEx);
            }
            // we no longer need the transaction handler
            fTransactionHandler = null;
            // we no longer need the client input stream!
            fClientInputStream = null;
            // we no longer need the client output stream!
            fClientOutputStream = null;
        } else {
            if (fClientOutputStream == null)
                System.err.println("fClientOutputStream null!");
            if (fClientInputStream == null)
                System.err.println("fClientInputStream null!");
        }

        if (fDebugOn) System.out.println("closing fClientSocket...");
        try {
            fClientSocket.close();
        } catch (Exception clientSocketCloseEx) {
            System.err.println("fClientSocket.close() threw: " +
                    clientSocketCloseEx);
        }

        fClientSocket = null;
        if (fDebugOn) System.out.println("done with cleanup...");
    }
}