

```

void downloadFile(const std::string& host, unsigned int port,
const std::string& filename)
{

    Lock* lock = acquireLock();
    Connection* connection = connect(host,port);
    File* file = openFile(filename)
    while(connection->hasMoreData())
    {
        file->writeLine(connection->readLine());
    }
    catch (// Whatever exception that user defined to throw)

}

int main()
{
    try
    {
        downloadFile(imhost.host, thatsport.port, hostporters);
    }
    catch (//some exception defined under assumptions)
    {

    }

    return 0;}

```

EXPLANATION::

It was much simpler than the Notes version. In the notes version i needed to think about all the ways it could go wrong and it forced me to keep writing the long cleanups in the catch that if i was writing a long program and had similiar functios i had to write a lot of cleanups.

RAII Helps to kind of abstract away some of the details you wrote for your own code (you abstracting your code away) thus helping maintaining a cleaner code that is MODULAR and if something goes wrong regarding the cleanup you don't have to look thru all the calls to the specific function and fix it but you will go to the approporaite header file and fix it there.