CPSC 327

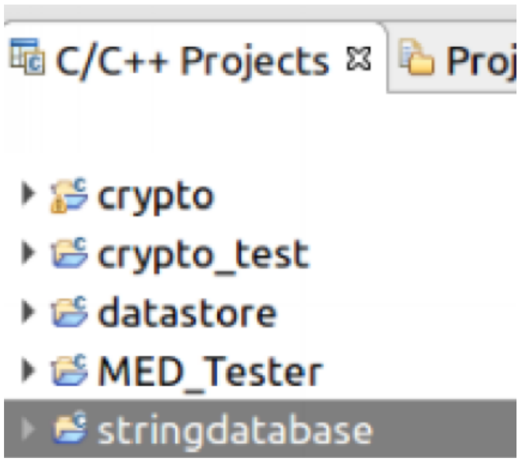
A Threadsafe, Serializable, Encrypting String Counter

Topics covered by this project;
Inheritance and Composition
Classes and Class Hierarchies
Concurrency and synchronization
Working with other codebases
A little encryption

Overview:
Design and develop a filestore object that saves and loads data to/from a file on disk. It must be able to encrypt/decrypt this data to/from disk.

Design and develop a threadsafe string counter. Threadsafe means it must handle multiple concurrent reads and writes while maintaining data integrity. It will use the above filestore to save and load this data from disk.

Opening Skeleton Projects
Here are the projects of interest, a description of each follows.



crypto - a Library (implemented for you) – links to nothing

A library that provides symmetric encryption services (same key encrypts and decrypts). Several classes. Ones to watch:

crypto: Abstract Base Class (ABC) for symmetric encryption algorithms (RC4, Triple DES, AES etc). See crypto.h for details and methods of interest

HexHelper static class used for encoding 1 byte binary to 2 byte hex. String doubles in size though, a better solution is base64 encoding.

Crypto_AES concrete class that derives from **Crypto**, encrypts and decrypts strings of data using a fully functional 128 bit AES encryption implementation.   It wraps a ReinDaal.cpp, a very complicated class (stay out its copied from the web and a somewhat ugly).

crypto_test – an application (implemented for you) – links to crypto library

An application that links to crypto and runs a simple test.

datastore – a Library (you implement some cpp files) – links to crypto library

A library that provides generic storage services. Intended to abstract details of where (file, network, DB etc. all the same to end user).

datastore: ABC for serialization. Also provides base encryption services using Crypto library above. See data_store.h for details and methods of interest.
Please develop the implementation (**data_store** .cpp file)

string_data: a storage class
A class with 2 members, a string and an int which counts the number of times the string was added. Header and cpp files provided for you.

data_store_file: loads and saves data to a file. Note the crypto base class pointer in the constructor. If not null then encrypt the data when saving and decrypt when loading.
Please develop the implementation (**data_store_file .cpp** file)

stringdatabase - a Library(you implement some cpp files) – links to datastore library

Main class that tracks the number of times individual strings occur. Holds string_data objects. **Must be threadsafe**. See string_database.h for api. You must use the DataStore pointer for loading and saving.
Please develop the implementation (**string_database .cpp** file)

MED_Tester – an Application (you implement) - links to crypto, datastore and stringdatabase libraries

Please develop the implementation (**MED_Tester.cpp file**). Test your libraries here, I will use my own version of **MED_Tester.cpp** that:

- Launches multiple threads that simultaneously read and write to the stringdatabase library.
- Serializes data to a file (with and without encryption)

Submission:

Only the following 4 files. Please do not zip them together, or embed them in a directory structure. Just the 4 files.

data_store .cpp
data_store_file .cpp
string_database .cpp
MED_Tester.cpp

Grading

I will use the following rubric to grade your project

10%	Files submitted in correct format
20%	Threadsafe operation
20%	Properly tracks strings
30%	Loads and saves correctly from/to a file no encryption.
30%	Loads and saves correctly from/to a file using encryption. Keep in mind that the data is not encrypted when it is stored in the vector, it is only encrypted when it is saved to disk and decrypted when it is loaded from disk.

Deliverables:

data_store .cpp, **data_store_file .cpp, string_database** .**cpp, MED_Tester.cpp.**

Questions:

What does string_database do on instantiation? Nothing it waits for the client app to either try to add a string to it or load using a data_store_file object.

Do I need to feed it particular types of strings? No any string will do as long as the database keeps track of how many times it has seen a distinct string.

How will you test my project?
Using my own test file

What if encryption only partly works?
The tester gives me a yes or a no. So it will be full or no credit.

What happens if it does not compile?
Nothing I can do here.

Will this look good on my Resume?

Probably, but you likely will have to know how the bits work. Here is a quick encryption overview.

AES is a symmetric key cipher algorithm. That is 1 key both encrypts and decrypts. This is what does the brunt of en/decryption.

The other major kind of encryption is asymmetric or public key. A public/private key pair is used. What the public key encrypts the private key decrypts and what the private key encrypts the public decrypts. You cannot however both encrypt and decrypt a document with the same key. They must be used in pairs. This kind of algorithm is used in key exchange (securely getting a symmetric encryption key from point to point) as well as digital signatures and digital certificates.