CS 405 Module Five Encryption Activity Guidelines and Rubric

Overview

You are a senior software developer on a team of software developers who are responsible for a large banking web application. Your manager has recently discovered that none of the data is encrypted. She is panicking and now wants to put encryption into everything. However, she wants some proof of how it works. Another developer on the team started creating the application to show encryption, but cannot figure out how to load and save files or to do encryption. The task has fallen to you to finish the work.

Prompt

You will learn to do the following:

- Implement XOR-based encryption character by character across a file of data
- Implement loading data from a text file into a string
- Save a string to a text file

The source code has been commented on using TODO to explain the detailed rules you must follow. When implementing XOR encryption, you will need to account for the length of your password key. See the example file in the module resources section. The file format that is loaded and saved is defined in the source code. You must use the format specified for the program to work. You are provided a sample input file to test your program, and you will need to change the data file to complete the assignment. Use the file names defined in the source code file, and don't forget that you can leverage capabilities provided by the standard C++ library to help you achieve success.

Specifically, you will need to complete the following:

- **Code XOR Encryption**: Complete the coding activity by successfully implementing XOR encryption, accounting for key lengths.
- **Code the Loading of a Text File**: Complete the coding activity by successfully implementing the logic to read a text file in the specified format into a string in the read_file method.
- **Code the Saving of a Text File**: Complete the coding activity by successfully implementing the logic to save a text file in the specified format in the save_data_file method.
- **Data Files**: Create an original data file in the specified data format, produce the encrypted and decrypted data files, and ensure all file names match the source code defined file names.
- **C/C++ Program Functionality and Best Practices**: Demonstrate industry standard best practices, including in-line comments and appropriate naming conventions to enhance readability of code. Develop functional C/C++ code that illustrates a software design pattern approach.
- **Process Summary**: Provide a summary of the debugging that is thorough and systematic, including specific types of bugs, and accurately describe the corrections.

To complete this assignment, download the Encryption files ZIP to use as guidance as you move through the activity. You will use your development environment to complete this activity.

What to Submit

Submit the following in a ZIP file: the completed source code, the original input data file, and the two data files generated by the application. Also include a Word document that contains a screenshot of the application console output and your brief process summary.

Module Five Encryption Activity Rubric

Criteria	Proficient (100%)	Needs Improvement (75%)	Not Evident (0%)	Value
Code XOR Encryption	Completes the coding activity by successfully implementing XOR encryption, accounting for key lengths	Completes most of the coding activity, but needs to address one or more of the following: accounting for key length, not encrypting the whole string, addressing math errors	Does not attempt criterion	20
Code the Loading of a Text File	Completes the coding activity by successfully implementing the logic to read a text file in the specified format into a string in the read_file method	Completes most of the coding activity, but does not read the data in the correct format	Does not attempt criterion	20
Code the Saving of a Text File	Completes the coding activity by successfully implementing the logic to save a text file in the specified format in the save_data_file method	Completes most of the coding activity, but does not save the files in the correct format	Does not attempt criterion	20

Data Files	Creates an original data file in the specified data format, produces the encrypted and decrypted data files, and ensures all file names match the source code defined file names	Provides some but not all three data files	Does not attempt criterion	20
C/C++ Program Functionality and Best Practices	Demonstrates industry standard best practices, including in-line comments and appropriate naming conventions to enhance readability of code; develops functional C/C++ code that illustrates a software design pattern approach	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include implementation of software design patterns in code	Does not attempt criterion	10
Process Summary	Provides a summary of the debugging that is thorough and systematic, including specific types of bugs, and accurately describes the corrections	Provides a summary of the debugging, but the summary lacks detail, the approach is unsystematic, specific types of bugs are not differentiated, or the corrections are not fully described	Does not attempt criterion	10
Total:				100%