5-2 Activity: Encryption Coding

A screenshot of a computer

AI-generated content may be incorrect.

For this project, I worked on a C++ program that shows how encryption works using files. The original program was incomplete and had several TODO sections that needed to be finished. The first thing I completed was the encryption function called encrypt\_decrypt(). I added XOR logic so that each character in the message gets scrambled using a repeating password. I used the modulus operator to cycle through the key if the message was longer than the key. This makes the message unreadable unless the correct password is used again to decrypt it, which is possible because XOR is reversible.

Next, I worked on reading the input file in the read\_file() function. I added code to open the file and load all its text into a single string using an input file stream and a string buffer. This allowed the entire content, including multiple lines and paragraphs, to be passed into the encryption function. After that, I completed the save\_data\_file() function, which is used to write either the encrypted or decrypted data to a new file. I added code to include the student’s name, the current date, the password , and a labeled section to mark where the actual message starts. I used plain text labels like “Student Name ->” and “<------ Original Data ------>” instead of fancy Unicode arrows to avoid display problems in different editors.

Finally, I tested the program by encrypting and then decrypting the file. I verified that the decrypted file matched the original input, and that the output files were clearly labeled and easy to understand. Each part I added was important to make sure the program worked from beginning to end, while also following good practices for file handling and encryption. Overall, this project taught me how to handle file input and output, use XOR encryption, and make program output easier to read and understand.