Intermediate-Level C++ Programming Assignment: File Handling with Encryption & Decryption

Objective:

The purpose of this assignment is to develop a solid understanding of file handling in C++. Students will implement a program that writes encrypted content into a text file and provides functionality to decrypt and display the content using a specific program.

Problem Statement:

You are required to design a C++ program that performs the following tasks:

1. **Write Encrypted Data to File:**
    - The user will enter a message, which will be encrypted and stored in a text file (`data.txt`).
    - The encryption method should be simple (e.g., Caesar Cipher with a fixed shift value or XOR encryption).
2. **Read and Decrypt Data from File:**
    - The program should provide an option to read the encrypted file, decrypt its contents, and display the original message on the console.
3. **Ensure Security:**
    - If a user opens the text file directly, they should see only encrypted content.
    - Only the provided C++ program should be able to decrypt and display the message.

Implementation Details:

1. **Create a class `FileEncryptor`** with the following members:
    - Private:

        - `std::string filename` - Stores the filename.
        - `int encryptionKey` - Fixed encryption key (e.g., shift value for Caesar Cipher or XOR key).
    - Public:
        - Constructor to initialize the filename and encryption key.
        - `void encryptAndWrite(std::string message)` - Encrypts the input message and writes it to a file.
        - `std::string readAndDecrypt()` - Reads encrypted content from the file and decrypts it.
        - `void displayFileContents()` - Displays decrypted content.
2. **Implement Encryption & Decryption:**
    - Use a simple encryption technique:
        - **Caesar Cipher**: Shift each character by a fixed number of positions.
        - **XOR Encryption**: Perform XOR operation on each character with a key.
3. **Demonstrate File Handling:**
    - Use `fstream` to write and read encrypted data.
    - Ensure proper error handling in case of missing files or read/write issues.

Example Output:

Enter a message to encrypt: Hello World
Message successfully encrypted and stored in data.txt

Reading and decrypting file...
Decrypted Message: Hello World

Contents of `data.txt` (if opened directly):

Khoor Zruog (for Caesar Cipher with shift 3)
OR
Some unreadable symbols (for XOR encryption)

Submission Requirements:

- Upload a single `.cpp` file containing the program.
- Ensure proper use of file handling and encryption techniques.
- Provide comments explaining key sections of the code.

Deadline: 14 April 2025