Credit
Extra?
Assignment
Security Suite

How has your program changed from planning to coding to now? Please explain?

```java
package securitysuite;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        CaesarCipher caesar = new CaesarCipher();
        VigenereCipher vigenere = new VigenereCipher();
        FileProcessor fileProcessor = new FileProcessor();

        while (true) {
            System.out.println("\n=== CipherGuard ===");
            System.out.println("[1] Caesar Encrypt Message");
            System.out.println("[2] Caesar Decrypt Message");
            System.out.println("[3] Encrypt File (Caesar)");
            System.out.println("[4] Brute Force Crack (Caesar)");
            System.out.println("[5] Vigenere Encrypt");
            System.out.println("[6] Vigenere Decrypt");
            System.out.println("[7] Exit");
            System.out.print("Choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // clear buffer

            switch (choice) {

                case 1: // Caesar Encrypt
                    System.out.print("Enter message: ");
                    String msg = scanner.nextLine();

                    System.out.print("Enter shift value: ");
                    int shift = scanner.nextInt();
                    scanner.nextLine();

                    String encrypted = caesar.encrypt(msg, shift);
                    System.out.println("Encrypted: " + encrypted);
                    break;

                case 2: // Caesar Decrypt
                    System.out.print("Enter encrypted message: ");
                    String encMsg = scanner.nextLine();

                    System.out.print("Enter shift value used: ");
                    int dShift = scanner.nextInt();
                    scanner.nextLine();

                    String decrypted = caesar.decrypt(encMsg, dShift);
                    System.out.println("Decrypted: " + decrypted);
                    break;

```

```
                    System.out.print("Enter shift value used: ");
                    int dShift = scanner.nextInt();
                    scanner.nextLine();

                    String decrypted = caesar.decrypt(encMsg, dShift);
                    System.out.println("Decrypted: " + decrypted);
                    break;

                case 3: // File Encrypt
                    System.out.print("Enter input file (.txt): ");
                    String input = scanner.nextLine();

                    System.out.print("Enter output file name: ");
                    String output = scanner.nextLine();

                    System.out.print("Enter shift value: ");
                    int fileShift = scanner.nextInt();
                    scanner.nextLine();

                    fileProcessor.encryptFile(input, output, fileShift);
                    break;

                case 4: // Brute force
                    System.out.print("Enter encrypted message: ");
                    String brute = scanner.nextLine();
                    fileProcessor.bruteForceCrack(brute);
                    break;

                case 5: // Vigenere Encrypt
                    System.out.print("Enter message: ");
                    String vMsg = scanner.nextLine();

                    System.out.print("Enter keyword: ");
                    String key = scanner.nextLine();

                    String vEncrypted = vigenere.encrypt(vMsg, key);
                    System.out.println("Encrypted: " + vEncrypted);
                    break;

                case 6: // Vigenere Decrypt
                    System.out.print("Enter encrypted message: ");
                    String vEnc = scanner.nextLine();

                    System.out.print("Enter keyword: ");
                    String vKey = scanner.nextLine();

                    String vDecrypted = vigenere.decrypt(vEnc, vKey);
                    System.out.println("Decrypted: " + vDecrypted);
                    break;

                case 7:
                    System.out.println("Exiting CipherGuard.");
                    scanner.close();
```

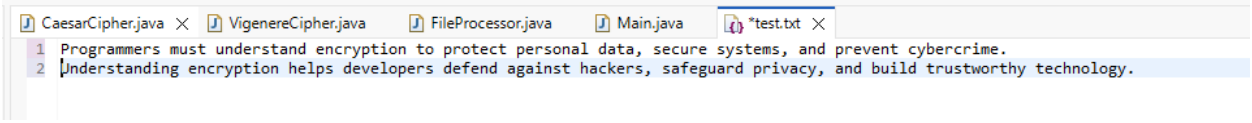
```
                    String vDecrypted = vigenere.decrypt(vEnc, vKey);
                    System.out.println("Decrypted: " + vDecrypted);
                    break;

                case 7:
                    System.out.println("Exiting CipherGuard.");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
```

```java
package securitysuite;

import java.io.*;

public class FileProcessor {

    private CaesarCipher cipher = new CaesarCipher();

    // Encrypt file from securitysuite package folder
    public void encryptFile(String inputFile, String outputFile, int shift) {

        // Force path to src/securitysuite folder
        String basePath = "src/securitysuite/";
        String inputPath = basePath + inputFile;
        String outputPath = basePath + outputFile;

        try (
            BufferedReader reader = new BufferedReader(new FileReader(inputPath));
            BufferedWriter writer = new BufferedWriter(new FileWriter(outputPath))
        ) {
            String line;

            while ((line = reader.readLine()) != null) {
                writer.write(cipher.encrypt(line, shift));
                writer.newLine();
            }

            System.out.println("File encrypted successfully → saved in securitysuite package");

        } catch (IOException e) {
            System.out.println("File error: " + e.getMessage());
            System.out.println("Make sure file is inside src/securitysuite/");
        }
    }

    // Brute force Caesar crack
    public void bruteForceCrack(String encryptedText) {
        long start = System.currentTimeMillis();

        System.out.println("\nBrute force results:\n");

        for (int shift = 0; shift < 26; shift++) {
            String attempt = cipher.decrypt(encryptedText, shift);
            System.out.println("Shift " + shift + ": " + attempt);
        }

        long duration = System.currentTimeMillis() - start;
        System.out.println("\nCompleted in: " + duration + " ms");
    }
}
```

```java
package securitysuite;

public class VigenereCipher {

    public String encrypt(String text, String keyword) {
        StringBuilder result = new StringBuilder();
        keyword = keyword.toLowerCase();

        int keyIndex = 0;

        for (char c : text.toCharArray()) {
            if (Character.isLetter(c)) {
                int shift = keyword.charAt(keyIndex % keyword.length()) - 'a';
                char base = Character.isUpperCase(c) ? 'A' : 'a';

                int newChar = (c - base + shift) % 26;
                c = (char) (newChar + base);

                keyIndex++;
            }
            result.append(c);
        }
        return result.toString();
    }

    public String decrypt(String text, String keyword) {
        StringBuilder result = new StringBuilder();
        keyword = keyword.toLowerCase();

        int keyIndex = 0;

        for (char c : text.toCharArray()) {
            if (Character.isLetter(c)) {
                int shift = keyword.charAt(keyIndex % keyword.length()) - 'a';
                char base = Character.isUpperCase(c) ? 'A' : 'a';

                int newChar = (c - base - shift) % 26;
                if (newChar < 0) {
                    newChar += 26;
                }

                c = (char) (newChar + base);
                keyIndex++;
            }
            result.append(c);
        }
        return result.toString();
    }
}
```

```java
package securitysuite;

public class CaesarCipher {

    public String encrypt(String text, int shift) {
        StringBuilder result = new StringBuilder();

        for (char c : text.toCharArray()) {
            if (Character.isLetter(c)) {
                char base = Character.isUpperCase(c) ? 'A' : 'a';

                int newChar = (c - base + shift) % 26;
                if (newChar < 0) {
                    newChar += 26;
                }

                c = (char) (newChar + base);
            }
            result.append(c);
        }
        return result.toString();
    }

    public String decrypt(String text, int shift) {
        return encrypt(text, -shift);
    }
}
```

Reflection and test file


CaesarCipher.java
VigenereCipher.java
FileProcessor.java
Main.java
*test.txt
1 Programmers must understand encryption to protect personal data, secure systems, and prevent cybercrime.
2 Understanding encryption helps developers defend against hackers, safeguard privacy, and build trustworthy technology.