Name: Jyothiradithya Neti

Contact Number: 9963831217

Email Id: jyothiradithya23@gmail.com

Assignment-2:

```
import java.util.Scanner; // Importing Scanner class for taking user input
// Main class definition
public class ceaser_cipher {
   public static void main(String[] args) {
      // Create a Scanner object to take input from user
      Scanner sc = new Scanner(System.in);
      // Boolean variable to control the while loop
      boolean val = true;
      // Loop runs continuously until user chooses to exit (option 3)
      while (val) {
         // Display the available menu options
         display();
         // Ask user to enter an option
         System.out.println("Enter the option (1.Encrypt, 2.Decrypt, 3.Exit): ");
         int s = sc.nextInt(); // Takes numeric option input
         sc.nextLine(); // Consumes the leftover newline character
         // Switch case based on user's choice
         switch (s) {
            case 1: // Case for encryption
               System.out.println("Enter the text to encrypt: ");
               String text = sc.nextLine(); // Read the full text (can include spaces)
               System.out.println("Enter the key (single digit): ");
               int key = sc.nextInt(); // Read the key (shift value)
```

```
    sc.nextLine(); // Consume leftover newline
    // Call the encryption function
    String encrypt = encrypted(text, key);
    // Display the encrypted result
    System.out.println("Encrypted String: " + encrypt);
    System.out.println("--------");
    break;
  case 2: // Case for decryption
    System.out.println("Enter the text to decrypt: ");
    String text1 = sc.nextLine(); // Read the encrypted text
    System.out.println("Enter the key (single digit): ");
    int key1 = sc.nextInt(); // Read the same key used for encryption
    sc.nextLine(); // Consume leftover newline
    // Call the encrypted text
    String encText = encrypted(text1, key1);
    // Call the decryption function
    String decrypt = decrypted(encText, key1);
    // Display the decrypted result
    System.out.println("Encrypted String: " + encText);
    System.out.println("Decrypted String: " + decrypt);
    System.out.println("--------");
    break;
  case 3: // Case to exit the program
    val = false; // Set loop variable to false → loop stops
    System.out.println("Exiting... Goodbye!");
    break;
  default: // Handles invalid option input
    System.out.println("Invalid option! Try again.");
```

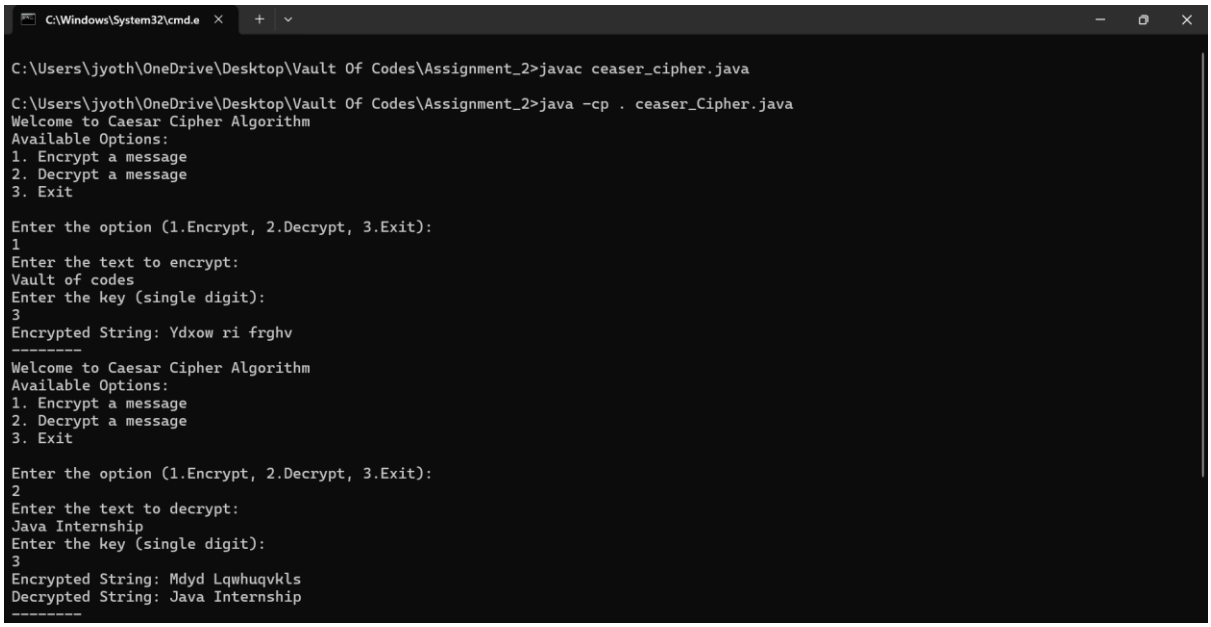
```
        }
    }
    // Close the scanner to free resources
    sc.close();
}

// ------------------- ENCRYPTION FUNCTION -------------------
public static String encrypted(String text, int key) {
    String result = ""; // String to store the final encrypted text
    // Loop through each character of the input text
    for (int i = 0; i < text.length(); i++) {
        char c = text.charAt(i); // Extract each character
        // If character is uppercase (A–Z)
        if (Character.isUpperCase(c)) {
            // Shift within A–Z range using modulo to wrap around after Z
            c = (char) (((c - 'A' + key) % 26) + 'A');
        }
        // If character is lowercase (a–z)
        else if (Character.isLowerCase(c)) {
            // Shift within a–z range using modulo
            c = (char) (((c - 'a' + key) % 26) + 'a');
        }
        // Non-alphabetic characters (spaces, digits, punctuation) remain unchanged
        result += c;
    }
    // Return the final encrypted string
    return result;
}
```

```
// -------------------- DECRYPTION FUNCTION --------------------
public static String decrypted(String text, int key) {
   String result = ""; // String to store final decrypted text
   // Loop through each character in the encrypted text
   for (int i = 0; i < text.length(); i++) {
      char c = text.charAt(i); // Extract character
      // If character is uppercase
      if (Character.isUpperCase(c)) {
         // Reverse the shift (subtract key) and wrap using modulo
         c = (char) (((c - 'A' - key + 26) % 26) + 'A');
      }
      // If character is lowercase
      else if (Character.isLowerCase(c)) {
         // Reverse shift for lowercase letters
         c = (char) (((c - 'a' - key + 26) % 26) + 'a');
      }
      // Non-letter characters remain the same
      result += c;
   }
   // Return the final decrypted text
   return result;
}
// -------------------- DISPLAY MENU FUNCTION --------------------
public static void display() {
   // This method displays the menu options every time the loop runs
   System.out.println("Welcome to Caesar Cipher Algorithm");
   System.out.println("Available Options:");
```

```
        System.out.println("1. Encrypt a message");
        System.out.println("2. Decrypt a message");
        System.out.println("3. Exit");
        System.out.println(); // Print a blank line for spacing
    }
}
```

Output:



Github Link: https://github.com/jyothiradithya-2345/Internship