Title: Caesar Cipher Encryption and Decryption.

Aims:

- Encrypt texts using Caesar Cipher word by word.
- Encrypt texts using Caesar Cipher as a whole sentence.
- Decryption using Caesar Cipher.

Tasks:

- Use Caesar Cipher method to encrypt words.
- Use Caesar Cipher method to encrypt sentences.
- Use Caesar Cipher method to decrypt Cipher texts.

Activities:

1. Use Caesar Cipher method to encrypt words.

```
package program1;
import java.util.*;

public class CeaserCipher_a {
	public static StringBuffer encrypt(String text, int s)
	{
		StringBuffer result= new StringBuffer();

		for (int i=0; i<text.length(); i++)
		{
			if (Character.isUpperCase(text.charAt(i)))
			{
				char ch = (char)(((int)text.charAt(i) + s - 65) % 26 + 65);
				result.append(ch);
			}
			else
			{
				char ch = (char)(((int)text.charAt(i) + s - 97) % 26 + 97);
				result.append(ch);
			}
		}
		return result;
	}
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the Text: ");
		String text = sc.nextLine();
		System.out.println("Enter the shift value: ");
		int s = sc.nextInt();
		System.out.println("Text : " + text);
		System.out.println("Shift : " + s);
		System.out.println("Cipher: " + encrypt(text, s));

	}
	}
```

2. Use Caesar Cipher method to encrypt sentences.

```
package program2;

import java.util.*;

public class CeaserCipher_b {
	public static void main(String args[]) {
		Scanner sc = new Scanner(System.in);
		System.out.println(" Input the plaintext message : ");
		String plaintext = sc.nextLine();
		System.out.println(" Enter the Key value: ");
		int shift = sc.nextInt();
		String ciphertext = "";
		char alphabet;
		for(int i=0; i < plaintext.length();i++)
		{
		     // Shift one character at a time
		    alphabet = plaintext.charAt(i);
		    // if alphabet lies between a and z
		    if(alphabet >= 'a' && alphabet <= 'z')
		    {
		     // shift alphabet
		     alphabet = (char) (alphabet + shift);
		     // if shift alphabet greater than 'z'
		     if(alphabet > 'z') {
		        // reshift to starting position
		        alphabet = (char) (alphabet+'a'-'z'-1);
		     }
		     ciphertext = ciphertext + alphabet;
		     //System.out.println(alphabet);
		    }

		    // if alphabet lies between 'A'and 'Z'
		    else if(alphabet >= 'A' && alphabet <= 'Z') {
		     // shift alphabet
		     alphabet = (char) (alphabet + shift);

		     // if shift alphabet greater than 'Z'
		     if(alphabet > 'Z') {
		         //reshift to starting position
		         alphabet = (char) (alphabet+'A'-'Z'-1);
		     }
		     ciphertext = ciphertext + alphabet;
		    }
		    else {
		     ciphertext = ciphertext + alphabet;
		    }

		  }
		  System.out.println(" ciphertext : " + ciphertext);
		 }
	}
```

3. Use Caesar Cipher method to decrypt Cipher texts

```
package program3;

import java.util.*;

public class CeaserCipher_Decryption {
       public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println(" Input the ciphertext message : ");
        String ciphertext = sc.nextLine();
        System.out.println(" Enter the shift value : ");
        int shift = sc.nextInt();
        String decryptMessage = "";
        char alphabet;
        for(int i=0; i < ciphertext.length();i++)
        {
            // Shift one character at a time
            alphabet = ciphertext.charAt(i);
            // if alphabet lies between a and z
            if(alphabet >= 'a' && alphabet <= 'z')
            {
                // shift alphabet
                alphabet = (char) (alphabet - shift);

                // shift alphabet lesser than 'a'
                if(alphabet < 'a') {
                    //reshift to starting position
                    alphabet = (char) (alphabet-'a'+'z'+1);
                }
                decryptMessage = decryptMessage + alphabet;
            }
                // if alphabet lies between A and Z
            else if(alphabet >= 'A' && alphabet <= 'Z')
            {
             // shift alphabet
                alphabet = (char) (alphabet - shift);

                //shift alphabet lesser than 'A'
                if (alphabet < 'A') {
                    // reshift to starting position
                    alphabet = (char) (alphabet-'A'+'Z'+1);
                }
                decryptMessage = decryptMessage + alphabet;
            }
            else
            {
             decryptMessage = decryptMessage + alphabet;
            }
        }
        System.out.println(" decrypt message : " + decryptMessage);
    }
}
```