**Practical 4 . Write a Java/C/C++/Python program to implement AES Algorithm**

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import java.util.Base64;

public class AESExample {

// Method to encrypt text using AES algorithm

public static String encrypt(String plainText, String secretKey) throws Exception {

// Create a SecretKeySpec object using the provided secret key

SecretKeySpec key = new SecretKeySpec(secretKey.getBytes(), "AES");

// Create a Cipher object for AES encryption

Cipher cipher = Cipher.getInstance("AES");

cipher.init(Cipher.ENCRYPT\_MODE, key);

// Encrypt the plaintext

byte[] encryptedBytes = cipher.doFinal(plainText.getBytes());

// Return the encrypted text in Base64 format

return Base64.getEncoder().encodeToString(encryptedBytes);

}

// Method to decrypt text using AES algorithm

public static String decrypt(String encryptedText, String secretKey) throws Exception {

// Create a SecretKeySpec object using the provided secret key

SecretKeySpec key = new SecretKeySpec(secretKey.getBytes(), "AES");

// Create a Cipher object for AES decryption

Cipher cipher = Cipher.getInstance("AES");

cipher.init(Cipher.DECRYPT\_MODE, key);

// Decode the encrypted text from Base64 format

byte[] decodedBytes = Base64.getDecoder().decode(encryptedText);

// Decrypt the encrypted bytes

byte[] decryptedBytes = cipher.doFinal(decodedBytes);

// Return the decrypted text

return new String(decryptedBytes);

}

public static void main(String[] args) {

try {

// Example secret key (16 bytes for AES-128)

String secretKey = "1234567890123456"; // 16-character key for AES-128 (must be 16 bytes long)

// Example plaintext

String plainText = "Hello, this is a test message!";

System.out.println("Original Text: " + plainText);

// Encrypt the plaintext

String encryptedText = encrypt(plainText, secretKey);

System.out.println("Encrypted Text (Base64): " + encryptedText);

// Decrypt the encrypted text

String decryptedText = decrypt(encryptedText, secretKey);

System.out.println("Decrypted Text: " + decryptedText);

} catch (Exception e) {

e.printStackTrace();

}

}

}