Practical-3 Write a Java/C/C++/Python program to implement DES algorithm.

```
import javax.crypto.Cipher;  // Import Cipher class for encryption/decryption
import javax.crypto.KeyGenerator; // Import KeyGenerator to generate secure DES keys
import javax.crypto.SecretKey;  // Import SecretKey interface for encryption keys
import javax.crypto.spec.SecretKeySpec; // Import SecretKeySpec to convert raw bytes to key
import java.util.Base64;  // Import Base64 for encoding/decoding encrypted text

public class DESAlgorithm {

    // Method to encrypt a message using DES
    public static String encrypt(String message, String key) throws Exception {
        // Create a SecretKeySpec using the provided key bytes and specify DES algorithm
        SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "DES");

        // Initialize the cipher with DES algorithm, using ECB mode and PKCS5 padding
        Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);  // Set cipher mode to encryption

        // Convert the message into bytes and encrypt it
        byte[] encryptedBytes = cipher.doFinal(message.getBytes());

        // Encode encrypted bytes to Base64 format and return it as a string
        return Base64.getEncoder().encodeToString(encryptedBytes);
    }

    // Method to decrypt a message using DES
    public static String decrypt(String encryptedMessage, String key) throws Exception {
```

```
    // Create a SecretKeySpec using the provided key bytes and specify DES algorithm
    SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "DES");

    // Initialize the cipher with DES algorithm, using ECB mode and PKCS5 padding
    Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
    cipher.init(Cipher.DECRYPT_MODE, secretKey);  // Set cipher mode to decryption

    // Decode the Base64 encoded encrypted message back to byte array
    byte[] decodedBytes = Base64.getDecoder().decode(encryptedMessage);

    // Decrypt the byte array back to original text
    byte[] decryptedBytes = cipher.doFinal(decodedBytes);

    // Convert decrypted byte array back to string and return
    return new String(decryptedBytes);
}

// Main method to test encryption and decryption
public static void main(String[] args) {
    try {
        // Define the plaintext message to be encrypted
        String message = "Hello, World!";

        // Define a fixed key (must be exactly 8 characters for DES)
        String key = "12345678"; // DES requires a key size of 8 bytes

        // Encrypt the message and store the encrypted result
        String encryptedMessage = encrypt(message, key);
```

```
            // Print the encrypted message to the console
            System.out.println("Encrypted Message: " + encryptedMessage);

            // Decrypt the encrypted message back to plaintext
            String decryptedMessage = decrypt(encryptedMessage, key);

            // Print the decrypted message to the console
            System.out.println("Decrypted Message: " + decryptedMessage);

        } catch (Exception e) {  // Catch any exceptions that may occur
            e.printStackTrace();  // Print error details if any exception occurs
        }
    }
}
```