❖ Write a Java program that allows users to create a digital signature for a piece of digital data provided by the user and then authenticate the digital signature.

The Implementation Java code is given below:

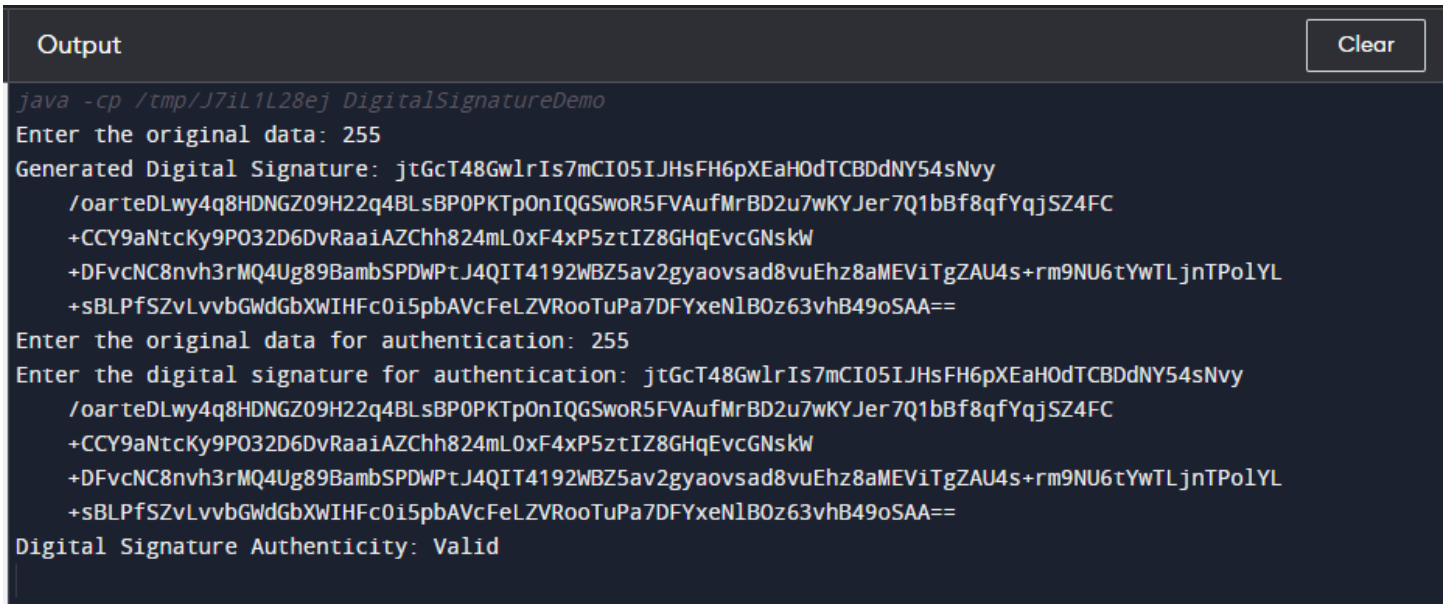
DigitalSignatureDemo.java

```java
import java.security.*;
import java.util.Base64;
import java.util.Scanner;
public class DigitalSignatureDemo {
    public static void main(String[] args) {
        try {
            // Step 1: Key Generation
            KeyPair keyPair = generateKeyPair();
            // Step 2: Create Digital Signature
            String originalData = getUserInput( prompt: "Enter the original data: ");
            byte[] digitalSignature = createDigitalSignature(originalData, keyPair.getPrivate());
            // Display the generated digital signature
            System.out.println("Generated Digital Signature: " + Base64.getEncoder().encodeToString(digitalSignature));
            // Step 3: Authenticate Digital Signature
            String inputData = getUserInput( prompt: "Enter the original data for authentication: ");
            String signatureInput = getUserInput( prompt: "Enter the digital signature for authentication: ");
            boolean isAuthentic = verifyDigitalSignature(inputData, Base64.getDecoder().decode(signatureInput), keyPair.getPublic());
            // Display the authentication result
            System.out.println("Digital Signature Authenticity: " + (isAuthentic ? "Valid" : "Invalid"));
        } catch (Exception e) {
            System.err.println("An error occurred: " + e.getMessage());
        }
    }
    1 usage
    private static KeyPair generateKeyPair() throws NoSuchAlgorithmException {
        KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance( algorithm: "RSA");
        keyPairGenerator.initialize( keysize: 2048); // You can adjust the key size
        return keyPairGenerator.generateKeyPair();
    }
    1 usage
    private static byte[] createDigitalSignature(String data, PrivateKey privateKey) throws NoSuchAlgorithmException, InvalidKeyException, SignatureException {
        Signature signature = Signature.getInstance( algorithm: "SHA256withRSA");
        signature.initSign(privateKey);
        signature.update(data.getBytes());
        return signature.sign();
    }
```

```java
    1 usage
    private static boolean verifyDigitalSignature(String data, byte[] signature, PublicKey publicKey)
            throws NoSuchAlgorithmException, InvalidKeyException, SignatureException {
        Signature verifier = Signature.getInstance( algorithm: "SHA256withRSA");
        verifier.initVerify(publicKey);
        verifier.update(data.getBytes());
        return verifier.verify(signature);
    }
    3 usages
    private static String getUserInput(String prompt) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(prompt);
        return scanner.nextLine();
    }
}
```

❖ **Output:**

1. **If data and signature is valid:** Here, the original data is 255 and the data is 260 for authentication. Because the data has been Same, the authenticity of the data is therefore valid.



2. **If one is invalid (data or signature):** Here, the original data is 255 and the data is 260 for authentication. Because the data has been altered, the authenticity of the data is therefore invalid.