Title: Encryption and Decryption.

Aims:

- Encrypting the texts.
- Decrypting the ciphertexts.

Tasks:

- Encrypt the plaintext using the Key Pair.
- Decrypting the respective ciphertexts using the Key Pair.

Activities:

1. Encrypt the plaintext using the Key Pair.

```
package signature_gen_verify;

import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.Signature;
import java.util.Scanner;

public class Signature_Gen {

   public static void main(String[] args) throws Exception{
          Scanner sc = new Scanner(System.in);
          System.out.println("Enter some text");
          String msg = sc.nextLine();

          KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");

          keyPairGen.initialize(2048);

          KeyPair pair = keyPairGen.generateKeyPair();

          PrivateKey privKey = pair.getPrivate();

          Signature sign = Signature.getInstance("SHA256withDSA");

          sign.initSign(privKey);
          byte[] bytes = "msg".getBytes();

          sign.update(bytes);

          byte[] signature = sign.sign();

          System.out.println("Digital signature for given text: "+new
String(signature, "UTF8"));
   }

}
```

2. Decrypting the respective ciphertexts using the Key Pair.

```
package signature_gen_verify;

import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.Signature;
import java.util.Scanner;

public class Verify_Sign {

   public static void main(String[] args) throws Exception{
          Scanner sc = new Scanner(System.in);
          System.out.println("Enter some text");
          String msg = sc.nextLine();

          KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");

          keyPairGen.initialize(2048);

          KeyPair pair = keyPairGen.generateKeyPair();

          PrivateKey privKey = pair.getPrivate();

          Signature sign = Signature.getInstance("SHA256withDSA");

          sign.initSign(privKey);
          byte[] bytes = "msg".getBytes();

          sign.update(bytes);

          byte[] signature = sign.sign();

          sign.initVerify(pair.getPublic());

          sign.update(bytes);

          boolean bool = sign.verify(signature);
          if(bool){
          System.out.println("Signature verified");
          }else{
          System.out.println("Signature failed");
   }
   }
}
```