**Name:** Aditya Somani **Roll No:** BE1851061 **PRN:** 71901204L

**Assignment Number:** 3

**Assignment Title:** Write a program in C++ or Java to implement SHA1 algorithm using libraries (API).

**Algorithm**

**Key Generation**

Select p, q

p and q both prime, p≠q

Calculate n = p \* q Calculate Ø(n) = (p-1)(q-1)

Select integer e gcd(Ø(n),e) = 1; 1<e<Ø(n)

Calculate d

d = e(-1) mod Ø(n)

Public key

PU = {e, n}

Private key

PR = {d, n}

**Encryption**

Plaintext

M<n

Ciphertext

C=Me mod n

**Decryption**

Ciphertext

C

Plaintext

M = Cd mod n

**Required Classes**

* + **KeyGeneration**

1. **Class KeyPairGenerator**

The KeyPairGenerator class is used to generate pairs of public and private keys. A Key pair generator for a particular algorithm creates a public/private key pair that can be used with this algorithm. It also associates algorithm-specific parameters with each of the generated keys.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| protected | [**KeyPairGenerator**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPairGenerator.html#KeyPairGenerator(java.lang.String))([String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) algorithm)  Creates a KeyPairGenerator object for the specified algorithm. |
| **Required Methods** | |
| [KeyPair](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPair.html) | [**generateKeyPair**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPairGenerator.html#generateKeyPair())()  Generates a key pair. |
| static [KeyPairGenerator](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPairGenerator.html) | [**getInstance**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPairGenerator.html#getInstance(java.lang.String))([String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) algorithm)  Generates a KeyPairGenerator object that implements the specified digest algorithm. |
| void | [**initialize**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPairGenerator.html#initialize(java.security.spec.AlgorithmParameterSpec%2C%20java.security.SecureRandom))([AlgorithmParameterSpec](http://download.oracle.com/javase/1.4.2/docs/api/java/security/spec/AlgorithmParameterSpec.html) params, [SecureRandom](http://download.oracle.com/javase/1.4.2/docs/api/java/security/SecureRandom.html) random)  Initializes the key pair generator with the given parameter set and source of randomness. |

## Class SecureRandom

This class provides a cryptographically strong pseudo-random number generator.

|  |
| --- |
| **Constructor Summary** |
| [**SecureRandom**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/SecureRandom.html#SecureRandom())()  By using this constructor, the caller obtains a SecureRandom object containing the implementation from the highest-priority installed provider that has a SecureRandom implementation. |
| [**SecureRandom**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/SecureRandom.html#SecureRandom(byte%5B%5D))(byte[] seed)  By using this constructor, the caller obtains a SecureRandom object containing |

|  |
| --- |
| the implementation from the highest-priority installed provider that has a SecureRandom implementation. |

## Class KeyPair

This class is a simple holder for a key pair (a public key and a private key). It does not enforce any security, and, when initialized, should be treated like a PrivateKey.

|  |  |  |
| --- | --- | --- |
| **Constructor Summary** | | |
| [**KeyPair**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPair.html#KeyPair(java.security.PublicKey%2C%20java.security.PrivateKey))([PublicKey](http://download.oracle.com/javase/1.4.2/docs/api/java/security/PublicKey.html) publicKey, [PrivateKey](http://download.oracle.com/javase/1.4.2/docs/api/java/security/PrivateKey.html) privateKey)  Constructs a key pair from the given public key and private key. | |  |
| **Required Methods** | | |
| [PrivateKey](http://download.oracle.com/javase/1.4.2/docs/api/java/security/PrivateKey.html) | [**getPrivate**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPair.html#getPrivate())()  Returns a reference to the private key component of this key  pair. | |
| [PublicKey](http://download.oracle.com/javase/1.4.2/docs/api/java/security/PublicKey.html) | [**getPublic**](http://download.oracle.com/javase/1.4.2/docs/api/java/security/KeyPair.html#getPublic())()  Returns a reference to the public key component of this key pair. | |

## Class ObjectOutputStream

An ObjectOutputStream writes primitive data types and graphs of Java objects to an OutputStream. The objects can be read (reconstituted) using an ObjectInputStream.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| protected | [**ObjectOutputStream**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectOutputStream.html#ObjectOutputStream())()  Provide a way for subclasses that are completely reimplementing ObjectOutputStream to not have to allocate private data just used by this implementation of ObjectOutputStream. |
|  | [**ObjectOutputStream**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectOutputStream.html#ObjectOutputStream(java.io.OutputStream))([OutputStream](http://download.oracle.com/javase/1.4.2/docs/api/java/io/OutputStream.html) out)  Creates an ObjectOutputStream that writes to the specified OutputStream. |
| **Required Methods** | |
| void | [**close**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectOutputStream.html#close())()  Closes the stream. |
| void | [**writeObject**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectOutputStream.html#writeObject(java.lang.Object))([Object](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/Object.html) obj) |

|  |  |
| --- | --- |
|  | Write the specified object to the ObjectOutputStream. |

## Encryption and Decryption

* 1. **Class KeyGenerator**

This class provides the functionality of a (symmetric) key generator.Key generators are constructed using one of the getInstance class methods of this class.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| protected | [**KeyGenerator**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/KeyGenerator.html#KeyGenerator(javax.crypto.KeyGeneratorSpi%2C%20java.security.Provider%2C%20java.lang.String))([KeyGeneratorSpi](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/KeyGeneratorSpi.html) keyGenSpi, [Provider](http://download.oracle.com/javase/1.4.2/docs/api/java/security/Provider.html) provider, [String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) algorithm)  Creates a KeyGenerator object. |
| **Required Methods** | |
| [SecretKey](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/SecretKey.html) | [**generateKey**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/KeyGenerator.html#generateKey())()  Generates a secret key. |
| static [KeyGenerator](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/KeyGenerator.html) | [**getInstance**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/KeyGenerator.html#getInstance(java.lang.String))([String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) algorithm)  Generates a KeyGenerator object for the specified algorithm. |
| void | [**init**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/KeyGenerator.html#init(java.security.SecureRandom))([SecureRandom](http://download.oracle.com/javase/1.4.2/docs/api/java/security/SecureRandom.html) random) Initializes this key generator. |

## Class ObjectInputStream

An ObjectInputStream deserializes primitive data and objects previously written using an ObjectOutputStream.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| protected | [**ObjectInputStream**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectInputStream.html#ObjectInputStream())()  Provide a way for subclasses that are completely reimplementing ObjectInputStream to not have to allocate private data just used by this implementation of ObjectInputStream. |
|  | [**ObjectInputStream**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectInputStream.html#ObjectInputStream(java.io.InputStream))([InputStream](http://download.oracle.com/javase/1.4.2/docs/api/java/io/InputStream.html) in) |

|  |  |
| --- | --- |
|  | Creates an ObjectInputStream that reads from the specified InputStream. |
| **Required Methods** | |
| void | [**close**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectInputStream.html#close())()  Closes the input stream. |
| [Object](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/Object.html) | [**readObject**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/ObjectInputStream.html#readObject())()  Read an object from the ObjectInputStream. |

## Interface Key

The Key interface is the top-level interface for all keys. It defines the functionality shared by all key objects.

## Class Cipher

This class provides the functionality of a cryptographic cipher for encryption and decryption.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| protected | [**Cipher**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#Cipher(javax.crypto.CipherSpi%2C%20java.security.Provider%2C%20java.lang.String))([CipherSpi](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/CipherSpi.html) cipherSpi, [Provider](http://download.oracle.com/javase/1.4.2/docs/api/java/security/Provider.html) provider, [String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) transformation)  Creates a Cipher object. |
| **Required Methods** | |
| byte[] | [**doFinal**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#doFinal())()  Finishes a multiple-part encryption or decryption operation, depending on how this cipher was initialized. |
| byte[] | [**doFinal**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#doFinal(byte%5B%5D%2C%20int%2C%20int))(byte[] input, int inputOffset, int inputLen)  Encrypts or decrypts data in a single-part operation, or finishes a multiple-part operation. |
| static [Cipher](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html) | [**getInstance**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#getInstance(java.lang.String))([String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) transformation)  Generates a Cipher object that implements the specified transformation. |
| void | [**init**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#init(int%2C%20java.security.Key))(int opmode, [Key](http://download.oracle.com/javase/1.4.2/docs/api/java/security/Key.html) key)  Initializes this cipher with a key. |
| [Key](http://download.oracle.com/javase/1.4.2/docs/api/java/security/Key.html) | [**unwrap**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#unwrap(byte%5B%5D%2C%20java.lang.String%2C%20int))(byte[] wrappedKey, [String](http://download.oracle.com/javase/1.4.2/docs/api/java/lang/String.html) wrappedKeyAlgorithm, int wrappedKeyType) |

|  |  |
| --- | --- |
|  | Unwrap a previously wrapped key. |
| byte[] | [**wrap**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#wrap(java.security.Key))([Key](http://download.oracle.com/javase/1.4.2/docs/api/java/security/Key.html) key)  Wrap a key. |
| int | [**update**](http://download.oracle.com/javase/1.4.2/docs/api/javax/crypto/Cipher.html#update(byte%5B%5D%2C%20int%2C%20int%2C%20byte%5B%5D))(byte[] input, int inputOffset, int inputLen, byte[] output)  Continues a multiple-part encryption or decryption operation (depending on how this cipher was initialized), processing another data part. |

## Class DataOutputStream

A data output stream lets an application write primitive Java data types to an output stream in a portable way.

|  |  |  |
| --- | --- | --- |
| **Constructor Summary** | | |
| [**DataOutputStream**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/DataOutputStream.html#DataOutputStream(java.io.OutputStream))([OutputStream](http://download.oracle.com/javase/1.4.2/docs/api/java/io/OutputStream.html) out)  Creates a new data output stream to write data to the specified underlying output stream. | |  |
| **Required Methods** | | |
| void | [**write**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/DataOutputStream.html#write(int))(int b)  Writes the specified byte (the low eight bits of the argument b) to the underlying output stream. | |
| void | [**writeInt**](http://download.oracle.com/javase/1.4.2/docs/api/java/io/DataOutputStream.html#writeInt(int))(int v)  Writes an int to the underlying output stream as four bytes, high byte first. | |

## Class InputStream

This abstract class is the superclass of all classes representing an input stream of bytes.

|  |  |
| --- | --- |
| **Required Methods** | |
| int | [**read**](http://download.oracle.com/javase/1.3/docs/api/java/io/InputStream.html#read(byte%5B%5D))(byte[] b)  Reads some number of bytes from the input stream and stores them into the buffer array b. |

## Class OutputStream

This abstract class is the superclass of all classes representing an output stream of bytes.

|  |  |
| --- | --- |
| **Required Methods** | |
| void | [**close**](http://download.oracle.com/javase/1.3/docs/api/java/io/OutputStream.html#close())()  Closes this output stream and releases any system resources associated with this stream. |
| void | [**write**](http://download.oracle.com/javase/1.3/docs/api/java/io/OutputStream.html#write(byte%5B%5D))(byte[] b)  Writes b.length bytes from the specified byte array to this output stream. |

## Algorithm

**Key Generation**

* + 1. Geta key pair generator object for generating keys for RSA algorithm by using KeyPairGenerator class.
    2. Initialize the Key Pair Generator.
    3. Generate the pair of keys by using KeyPair class.
    4. Write public and private keys to two different files.

## Encryption

1. Initialize cipher in encrypt mode.
2. Read data from plaintext file into byte array.
3. Call doFinal method to carry out encryption operation.
4. Write output array to ciphertext file.

## Decryption

1. Initialize cipher in decrypt mode.
2. Read data from ciphertext file into byte array.
3. Call doFinal method to carry out encryption operation.
4. Write output array to decrypted file.

**Code:**

import java.io.ByteArrayInputStream;

import java.io.ByteArrayOutputStream;

import java.security.DigestInputStream;

import java.security.DigestOutputStream;

import java.security.MessageDigest;

import java.security.Security;

import java.io.\*;

// Basic IO example using SHA1

public class SHA1

{

public static void main(String[] args) throws Exception

{

FileInputStream fstream = new FileInputStream("textfile.txt");

DataInputStream in = new DataInputStream(fstream);

BufferedReader br = new BufferedReader(new InputStreamReader(in));

String strLine;

strLine = br.readLine();

byte[] input = strLine.getBytes();

System.out.println("input : " + new String(input));

MessageDigest hash = MessageDigest.getInstance("SHA1");

ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(input);

DigestInputStream digestInputStream = new DigestInputStream(byteArrayInputStream, hash);

ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();

int ch;

while ((ch = digestInputStream.read()) >= 0) {

byteArrayOutputStream.write(ch);

}

byte[] newInput = byteArrayOutputStream.toByteArray();

System.out.println("in digest : " + new String(digestInputStream.getMessageDigest().digest()));

byteArrayOutputStream = new ByteArrayOutputStream();

DigestOutputStream digestOutputStream = new DigestOutputStream(byteArrayOutputStream, hash);

digestOutputStream.write(newInput);

digestOutputStream.close();

System.out.println("out digest: " + new String(digestOutputStream.getMessageDigest().digest()));

}

}

**Textfile.txt:**

Hello world, Good Morning

**Output:**

input : Hello world, Good Morning

in digest : d?▒×▒▒▒Xv]▒▒õ

out digest: d?▒×▒▒▒Xv]▒▒õ