Anthony Constantino
HW 7

Library
Java Cryptography Extension
http://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html

ECB encrypt/decrypt

```
public static byte[] encrypt(String plainText, String encryptionKey) throws Exception
{
	Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
	SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
	cipher.init(Cipher.ENCRYPT_MODE, key);
	return cipher.doFinal(plainText.getBytes("UTF-8"));
}

public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception
{
	Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
	SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
	cipher.init(Cipher.DECRYPT_MODE, key);
	return new String(cipher.doFinal(cipherText),"UTF-8");
}
```

CBC encrypt/decrypt

```
public static byte[] encrypt(String plainText, String encryptionKey) throws Exception
{
	Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding", "SunJCE");
	SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
	cipher.init(Cipher.ENCRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
	return cipher.doFinal(plainText.getBytes("UTF-8"));
}

public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception
{
	Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding", "SunJCE");
	SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
	cipher.init(Cipher.DECRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
	return new String(cipher.doFinal(cipherText),"UTF-8");
}
```

IV
AAAAAAAAAAAAAAAA

Plaintext
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Encryption Key
0123456789abcdef

ECB Ciphertext
64D87547CF781A845AA4B1E907E8205164D87547CF781A845AA4B1E907E8205164D87547CF781A845AA4B
1E907E8205164D87547CF781A845AA4B1E907E820518313BE1508EA3AB2E5B10E44A04986D4

CBC Ciphertext
BC264F3F987BBC37422BDC899692F23212418E7953B3F7D82BF3950F6A0500ABFC52214402DEAF5C3D078
04932535AF05B480A85EB71ABB0518A941B42A91F314C064C8CF59966957C01118B22302127

Lessons Learned

If the chosen coding language is well used there is most likely libraries out there for handling the encryption/decryption.

Java has libraries for everything from AES to Diffie-Hellman.

Sometimes documentation can be hairy to go through.

Do research on packages before using them in production.