Title: Keys and KeyStore.

Aims:

- Storing keys.
- Retrieving the stored keys.

Tasks:

- Create a KeyStore object, load it and store it.
- Retrieve the stored KeyStore object from the file.

Activities:

1. Create a KeyStore object, load it and Store it.

```
package storeRetrieveKeys;

import java.io.FileInputStream;
import java.security.KeyStore;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

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

         KeyStore keyStore = KeyStore.getInstance("JCEKS");

         char[] password = "changeit".toCharArray();
         String path = "C:\\Program
Files\\Java\\jre1.8.0_281\\lib\\security\\cacerts";
         java.io.FileInputStream fis = new FileInputStream(path);
         keyStore.load(fis, password);

         KeyStore.ProtectionParameter protectionParam = new
KeyStore.PasswordProtection(password);

         SecretKey mySecretKey = new SecretKeySpec("myPassword".getBytes(),
"DSA");

         KeyStore.SecretKeyEntry secretKeyEntry = new
KeyStore.SecretKeyEntry(mySecretKey);
         keyStore.setEntry("secretKeyAlias", secretKeyEntry, protectionParam);

         java.io.FileOutputStream fos = null;
         fos = new java.io.FileOutputStream("newKeyStoreName");
         keyStore.store(fos, password);
         System.out.println("data stored");
   }
}  }

}
```

2. Retrieve the stored KeyStore object from the file.

```
package storeRetrieveKeys;

import java.io.FileInputStream;
import java.security.KeyStore;
import java.security.KeyStore.ProtectionParameter;
import java.security.KeyStore.SecretKeyEntry;

import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

public class RetrieveKey {

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

          KeyStore keyStore = KeyStore.getInstance("JCEKS");

          char[] password = "changeit".toCharArray();
          java.io.FileInputStream fis = new FileInputStream("C:\\Program
Files\\Java\\jre1.8.0_281\\lib\\security\\cacerts");
          keyStore.load(fis, password);

          ProtectionParameter protectionParam = new
KeyStore.PasswordProtection(password);

          SecretKey mySecretKey = new SecretKeySpec("myPassword".getBytes(),
"DSA");

          SecretKeyEntry secretKeyEntry = new SecretKeyEntry(mySecretKey);
          keyStore.setEntry("secretKeyAlias", secretKeyEntry, protectionParam);

          java.io.FileOutputStream fos = null;
          fos = new java.io.FileOutputStream("newKeyStoreName");
          keyStore.store(fos, password);

          SecretKeyEntry secretKeyEnt =
(SecretKeyEntry)keyStore.getEntry("secretKeyAlias", protectionParam);

          SecretKey mysecretKey = secretKeyEnt.getSecretKey();
          System.out.println("Algorithm used to generate key :
"+mysecretKey.getAlgorithm());
          System.out.println("Format used for the key: "+mysecretKey.getFormat());
   }

}
```