PROGRAM:

KeyStore

Command to create KeyStore file:

```
keytool -genkeypair -keyalg RSA -keysize 2048 -validity 365 -alias myserverkey -keystore
samlKeystore.jks -storepass password -keypass password -dname
"CN=localhost,OU=Unknown,O=Unknown,L=Unknown,ST=Unknown,C=Unknown"
```

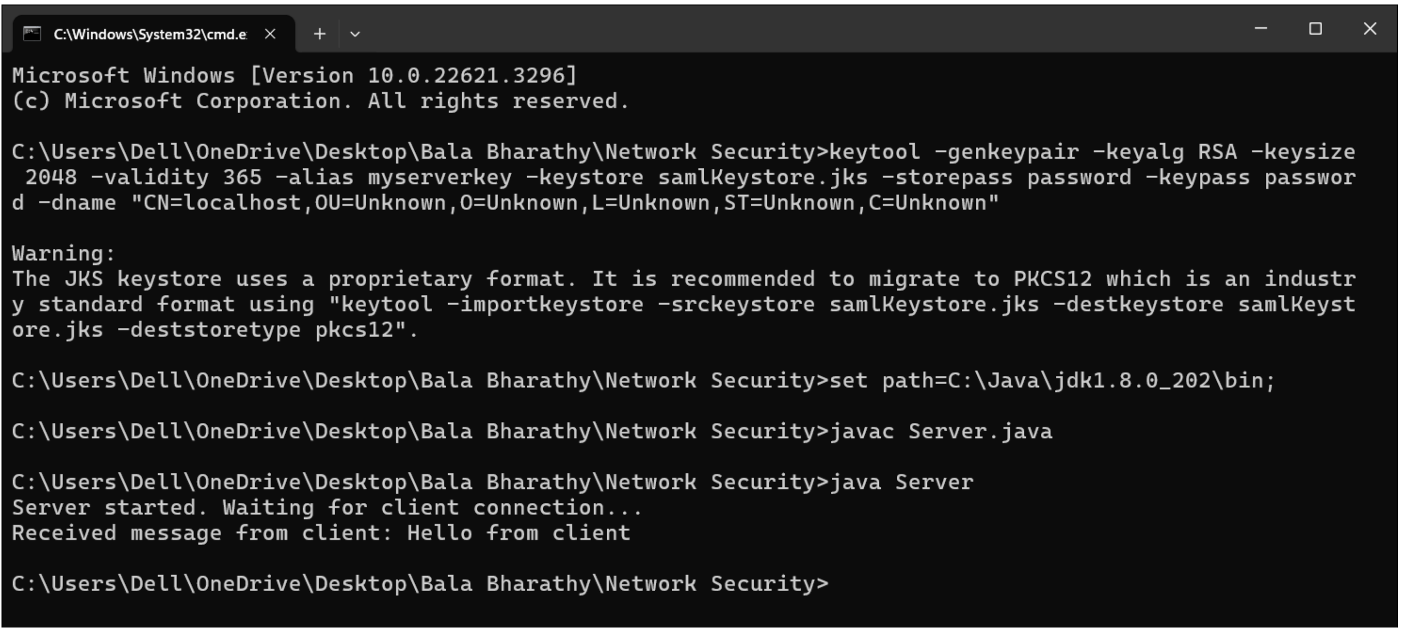
Server.java

```java
import javax.net.ssl.*;
import java.io.*;
import java.security.*;

public class Server {
    public static void main(String[] args) {
        try {
            // Load the keystore
            char[] keystorePassword = "password".toCharArray();
            char[] keyPassword = "password".toCharArray();
            KeyStore keyStore = KeyStore.getInstance("JKS");
            try (FileInputStream fis = new FileInputStream("samlKeystore.jks")) {
                keyStore.load(fis, keystorePassword);
            }

            // Set up the key manager factory
            KeyManagerFactory kmf = KeyManagerFactory.getInstance("SunX509");
            kmf.init(keyStore, keyPassword);

            // Set up the SSL context
            SSLContext sslContext = SSLContext.getInstance("TLS");
            sslContext.init(kmf.getKeyManagers(), null, null);

            // Create the server socket factory
            SSLServerSocketFactory ssf = sslContext.getServerSocketFactory();
            SSLServerSocket serverSocket = (SSLServerSocket) ssf.createServerSocket(9999);

            System.out.println("Server started. Waiting for client connection...");

            // Accept client connections
            SSLSocket socket = (SSLSocket) serverSocket.accept();

            // Set up input and output streams
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            // Read message from client
            String message = in.readLine();
            System.out.println("Received message from client: " + message);

            // Send response back to client
            out.println("Message received by server");

            // Close streams and socket
            out.close();
            in.close();
            socket.close();
            serverSocket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

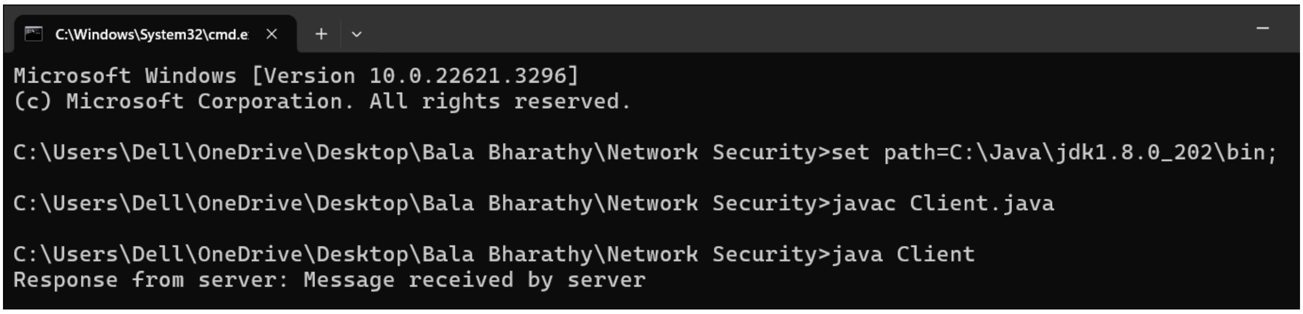
Client.java

```
import javax.net.ssl.*;
import java.io.*;
import java.security.*;

public class Client {
    public static void main(String[] args) throws Exception {
        // Load the truststore
        char[] truststorePassword = "password".toCharArray();
        KeyStore trustStore = KeyStore.getInstance("JKS");
        FileInputStream fis = new FileInputStream("samlKeystore.jks");
        trustStore.load(fis, truststorePassword);

        // Set up the trust manager factory
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509");
        tmf.init(trustStore);

        // Set up the SSL context
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the socket factory
        SSLSocketFactory sf = sslContext.getSocketFactory();
        SSLSocket socket = (SSLSocket) sf.createSocket("localhost", 9999);

        // Set up input and output streams
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

        // Send message to server
        out.println("Hello from client");

        // Read response from server
        String response = in.readLine();
        System.out.println("Response from server: " + response);

        // Close streams and socket
        out.close();
        in.close();
        socket.close();
    }
}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>keytool -genkeypair -keyalg RSA -keysize
 2048 -validity 365 -alias myserverkey -keystore samlKeystore.jks -storepass password -keypass passwor
d -dname "CN=localhost,OU=Unknown,O=Unknown,L=Unknown,ST=Unknown,C=Unknown"

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industr
y standard format using "keytool -importkeystore -srckeystore samlKeystore.jks -destkeystore samlKeyst
ore.jks -deststoretype pkcs12".

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>set path=C:\Java\jdk1.8.0_202\bin;

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>javac Server.java

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>java Server
Server started. Waiting for client connection...
Received message from client: Hello from client

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>
```



```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>set path=C:\Java\jdk1.8.0_202\bin;

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>javac Client.java

C:\Users\Dell\OneDrive\Desktop\Bala Bharathy\Network Security>java Client
Response from server: Message received by server
```