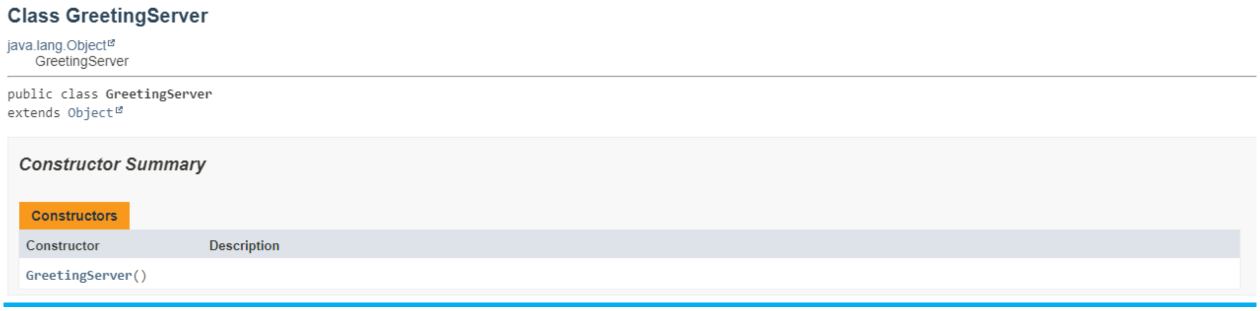
Documentation: IT Security Programming Task

The Diffie-Hellman key exchange algorithm combined with AES symmetric encryption between Server & Client

Amin Hassairi

Server source code description:



Class GreetingServer

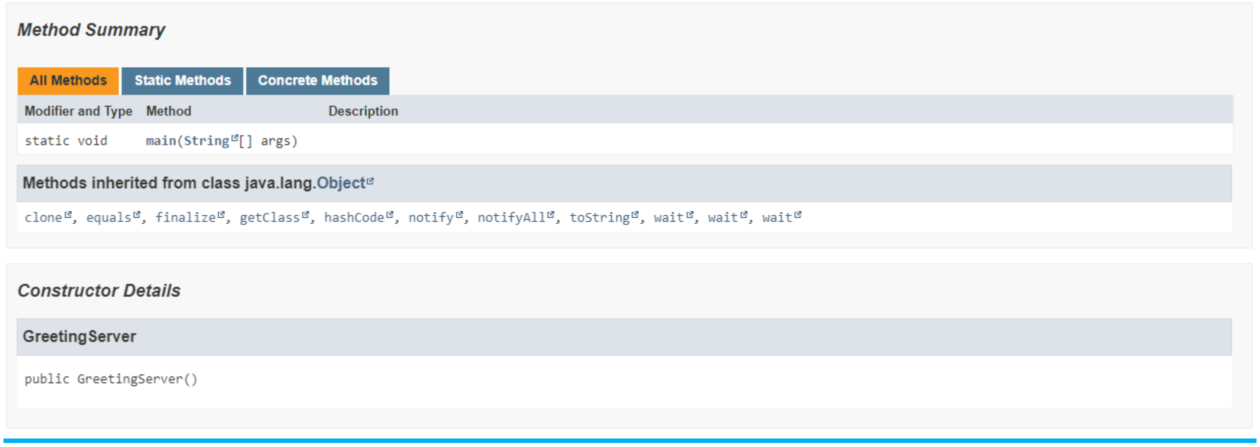
java.lang.Object
    GreetingServer

```
public class GreetingServer
extends Object
```

Constructor Summary

Constructors

Constructor	Description
GreetingServer()	



Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method	Description
static void	main(String[] args)	

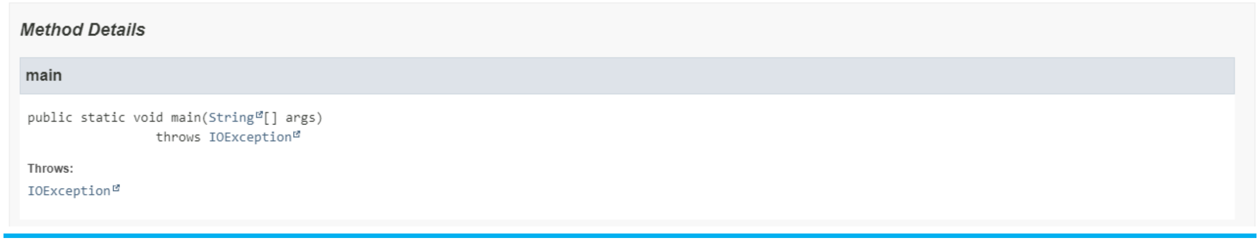
Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

GreetingServer

```
public GreetingServer()
```



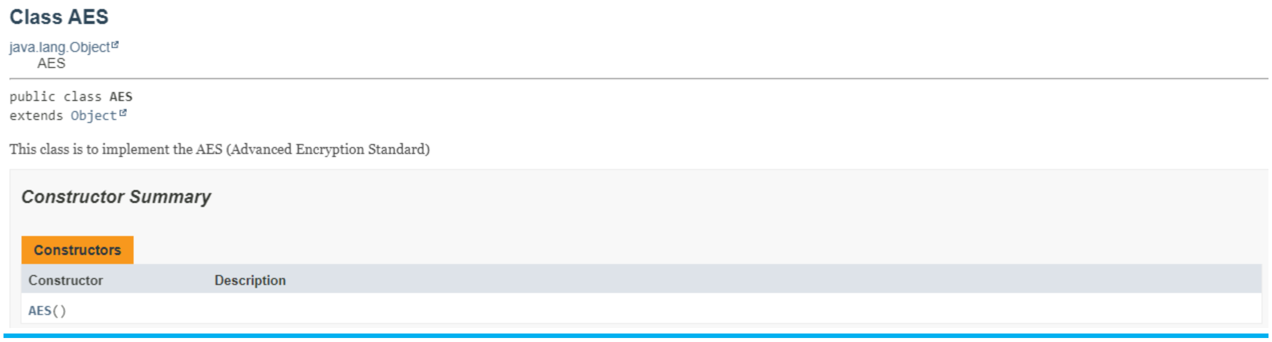
Method Details

main

```
public static void main(String[] args)
                 throws IOException
```

Throws:

IOException



Class AES

java.lang.Object
    AES

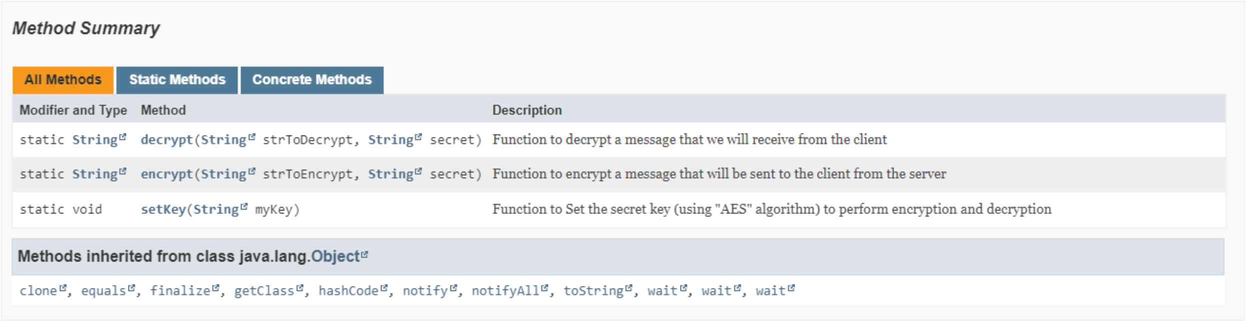
```
public class AES
extends Object
```

This class is to implement the AES (Advanced Encryption Standard)

Constructor Summary

Constructors

Constructor	Description
AES()	

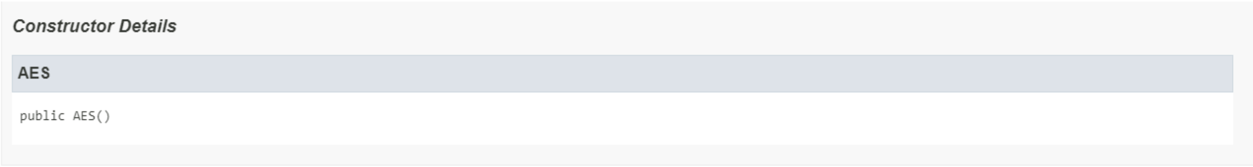


Method Summary

Modifier and Type	Method	Description
static String	decrypt(String strToDecrypt, String secret)	Function to decrypt a message that we will receive from the client
static String	encrypt(String strToEncrypt, String secret)	Function to encrypt a message that will be sent to the client from the server
static void	setKey(String myKey)	Function to Set the secret key (using "AES" algorithm) to perform encryption and decryption

Methods inherited from class java.lang.Object

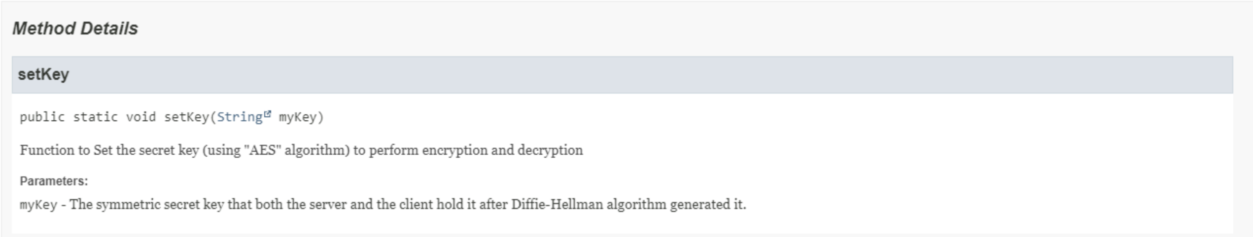
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`



Constructor Details

AES

```
public AES()
```



Method Details

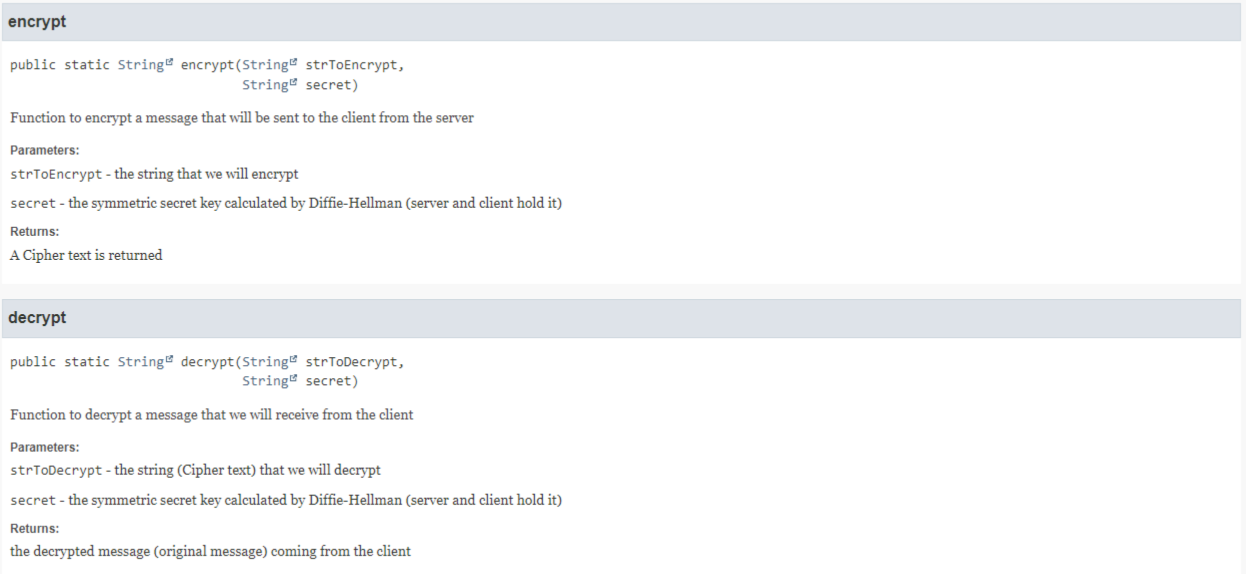
setKey

```
public static void setKey(String myKey)
```

Function to Set the secret key (using "AES" algorithm) to perform encryption and decryption

Parameters:

`myKey` - The symmetric secret key that both the server and the client hold it after Diffie-Hellman algorithm generated it.



encrypt

```
public static String encrypt(String strToEncrypt,
                             String secret)
```

Function to encrypt a message that will be sent to the client from the server

Parameters:

`strToEncrypt` - the string that we will encrypt

`secret` - the symmetric secret key calculated by Diffie-Hellman (server and client hold it)

Returns:

A Cipher text is returned

decrypt

```
public static String decrypt(String strToDecrypt,
                             String secret)
```

Function to decrypt a message that we will receive from the client

Parameters:

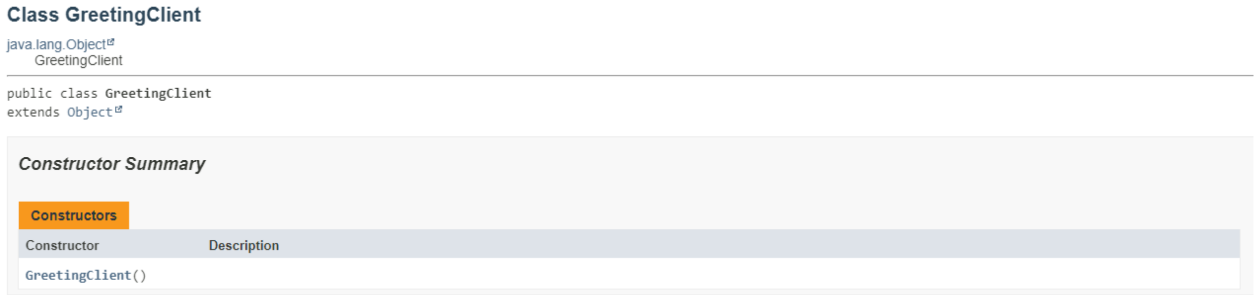
`strToDecrypt` - the string (Cipher text) that we will decrypt

`secret` - the symmetric secret key calculated by Diffie-Hellman (server and client hold it)

Returns:

the decrypted message (original message) coming from the client

Client source code description:



Class GreetingClient

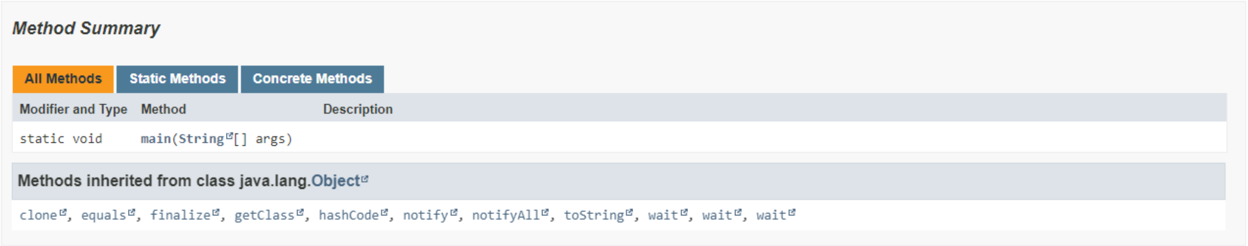
java.lang.Object
    GreetingClient

```
public class GreetingClient
extends Object
```

Constructor Summary

Constructors

Constructor	Description
`GreetingClient()`	



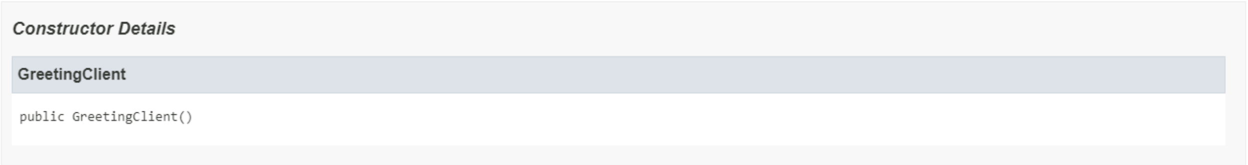
Method Summary

All Methods **Static Methods** **Concrete Methods**

Modifier and Type	Method	Description
`static void`	`main(String[] args)`	

Methods inherited from class java.lang.Object

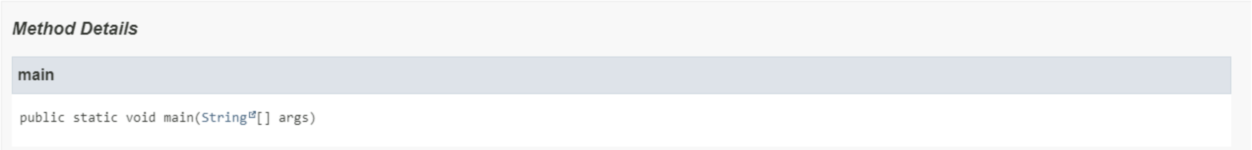
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`



Constructor Details

GreetingClient

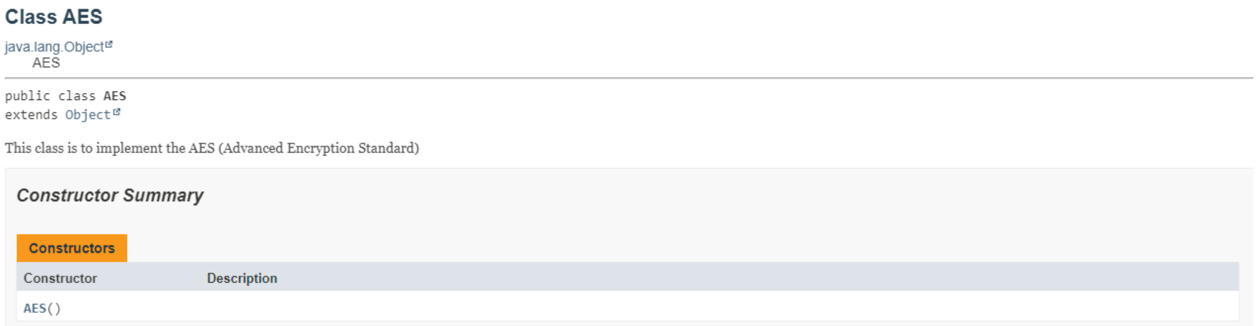
```
public GreetingClient()
```



Method Details

main

```
public static void main(String[] args)
```



Class AES

java.lang.Object
    AES

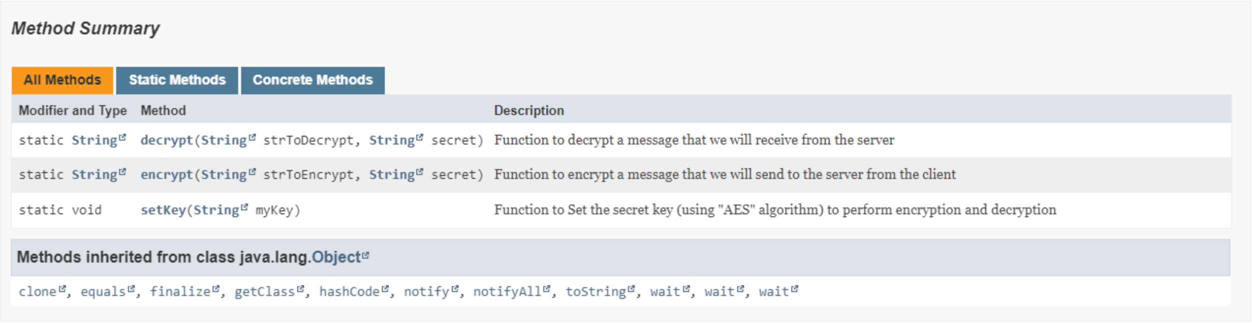
```
public class AES
extends Object
```

This class is to implement the AES (Advanced Encryption Standard)

Constructor Summary

Constructors

Constructor	Description
`AES()`	

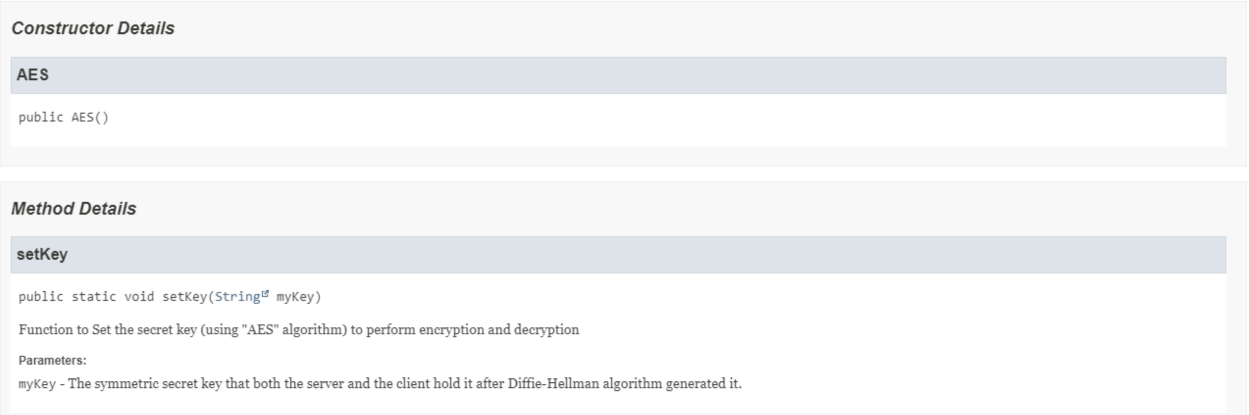


Method Summary

Modifier and Type	Method	Description
`static String`	`decrypt(String strToDecrypt, String secret)`	Function to decrypt a message that we will receive from the server
`static String`	`encrypt(String strToEncrypt, String secret)`	Function to encrypt a message that we will send to the server from the client
`static void`	`setKey(String myKey)`	Function to Set the secret key (using "AES" algorithm) to perform encryption and decryption

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`



Constructor Details

AES

```
public AES()
```

Method Details

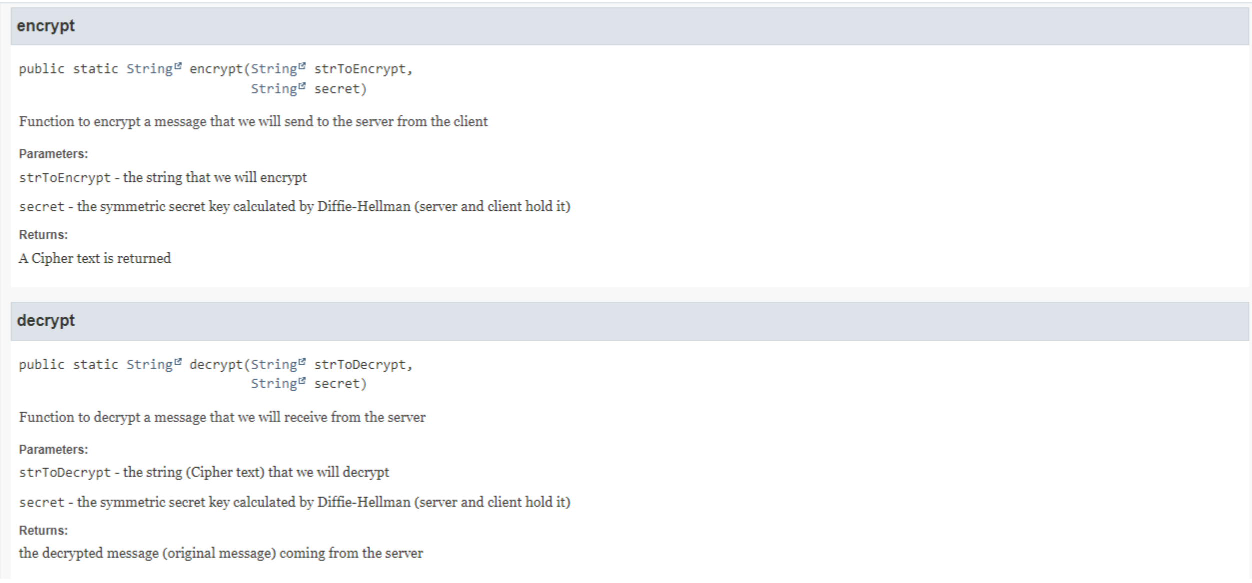
setKey

```
public static void setKey(String myKey)
```

Function to Set the secret key (using "AES" algorithm) to perform encryption and decryption

Parameters:

`myKey` - The symmetric secret key that both the server and the client hold it after Diffie-Hellman algorithm generated it.



encrypt

```
public static String encrypt(String strToEncrypt,
                             String secret)
```

Function to encrypt a message that we will send to the server from the client

Parameters:

`strToEncrypt` - the string that we will encrypt

`secret` - the symmetric secret key calculated by Diffie-Hellman (server and client hold it)

Returns:

A Cipher text is returned

decrypt

```
public static String decrypt(String strToDecrypt,
                             String secret)
```

Function to decrypt a message that we will receive from the server

Parameters:

`strToDecrypt` - the string (Cipher text) that we will decrypt

`secret` - the symmetric secret key calculated by Diffie-Hellman (server and client hold it)

Returns:

the decrypted message (original message) coming from the server

Running environment:

*Using IntelliJ IDEA Java for the coding solution.

*JDK 17.

*Localhost / Apache Tomcat.

*CMD command also available after we have used JAVAC.

Configuration files:

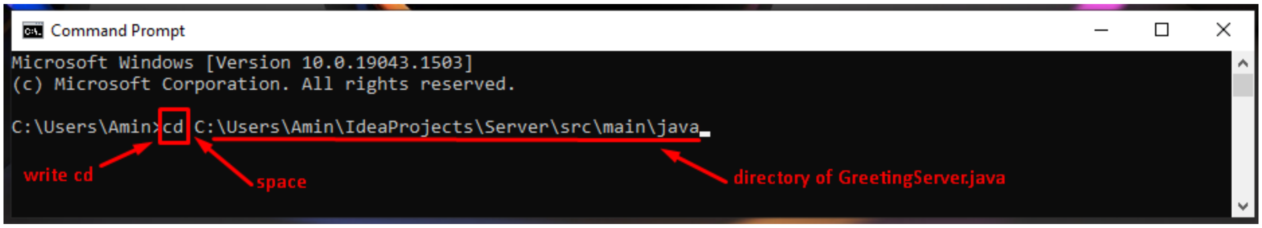
*After unzipping the file you will find two files: one called Server and the other Client.

*Open CMD command.

*Locate the Server file location and follow the steps shown on the next screenshots.

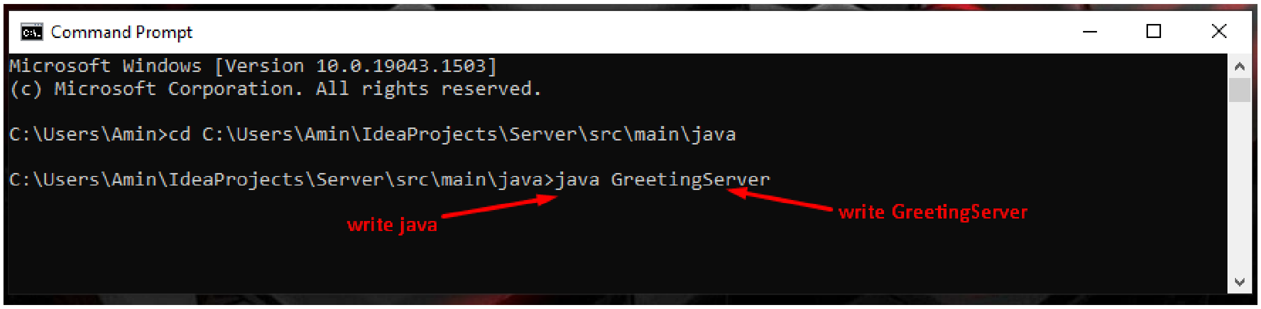
.\Server\src\main\java

*Write: cd {directory of GreetingServer.java} and hit enter.



```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Server\src\main\java
```

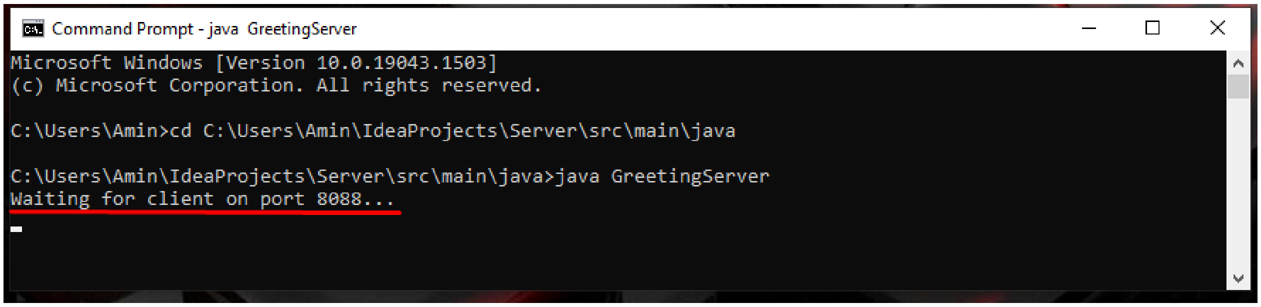
*Write: java GreetingServer and hit enter as shown below.



```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Server\src\main\java

C:\Users\Amin\IdeaProjects\Server\src\main\java>java GreetingServer
```

Amin Hassairi

*Now the Server is executed and here is the result that we can observe: The server is waiting (Listening) for the client on port 8088.



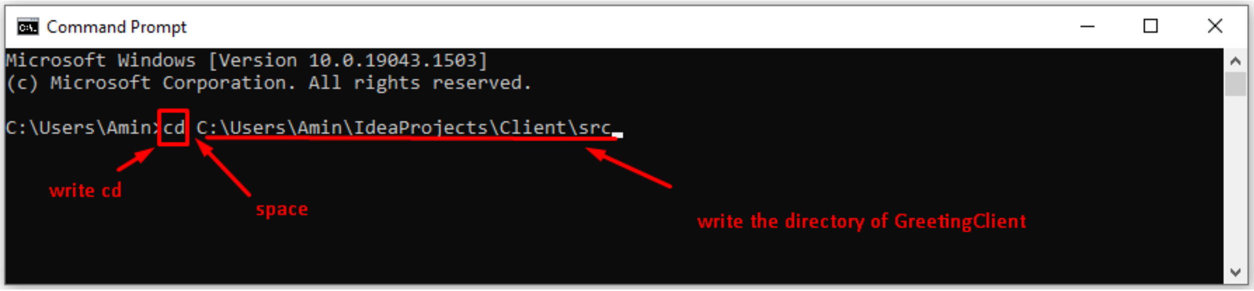
```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Server\src\main\java

C:\Users\Amin\IdeaProjects\Server\src\main\java>java GreetingServer
Waiting for client on port 8088...
```

*Open another CMD command.

*Locate the Client file location and follow the steps shown on the next screenshots.

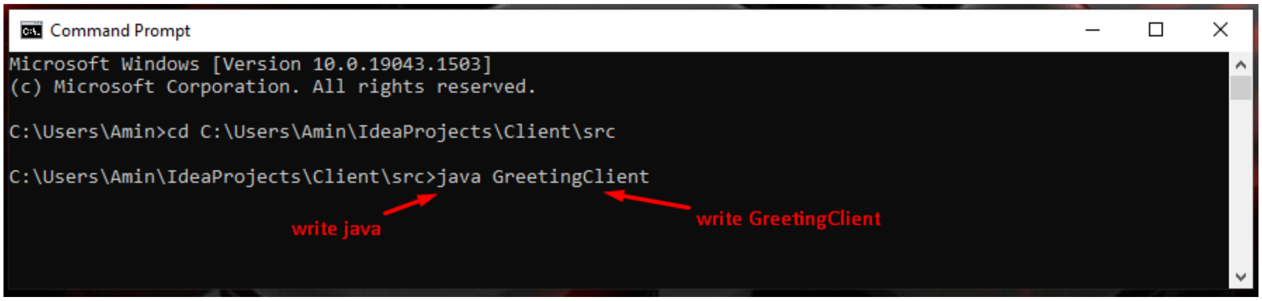
.\Client\src\main\java

*Write: cd {directory of GreetingClient.java} and hit enter.



```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Client\src
```

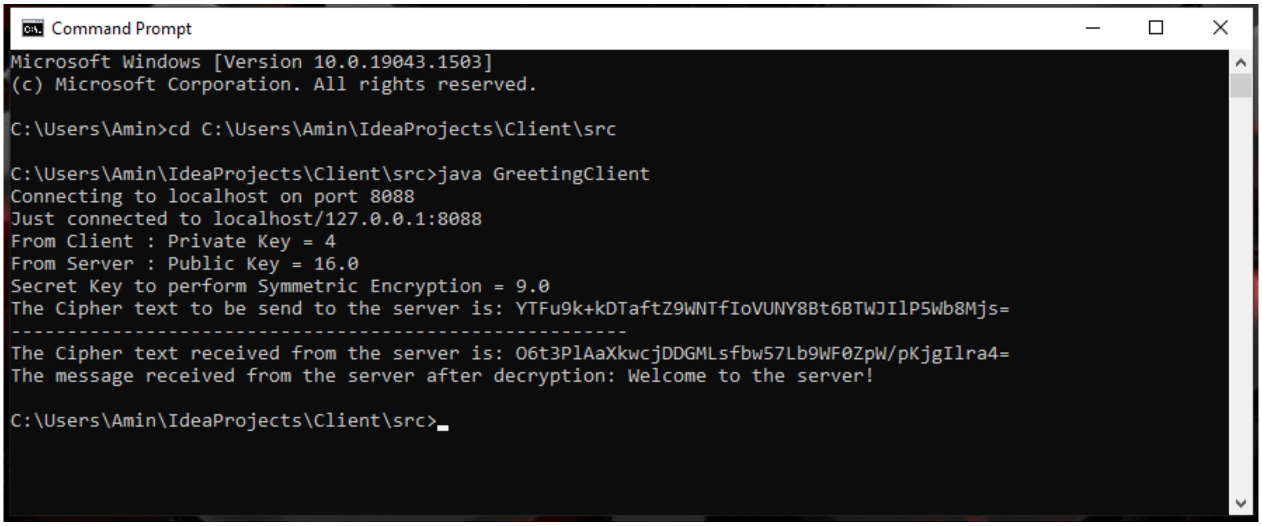
*Write: java GreetingClient and hit enter as shown below.



```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Client\src

C:\Users\Amin\IdeaProjects\Client\src>java GreetingClient
```

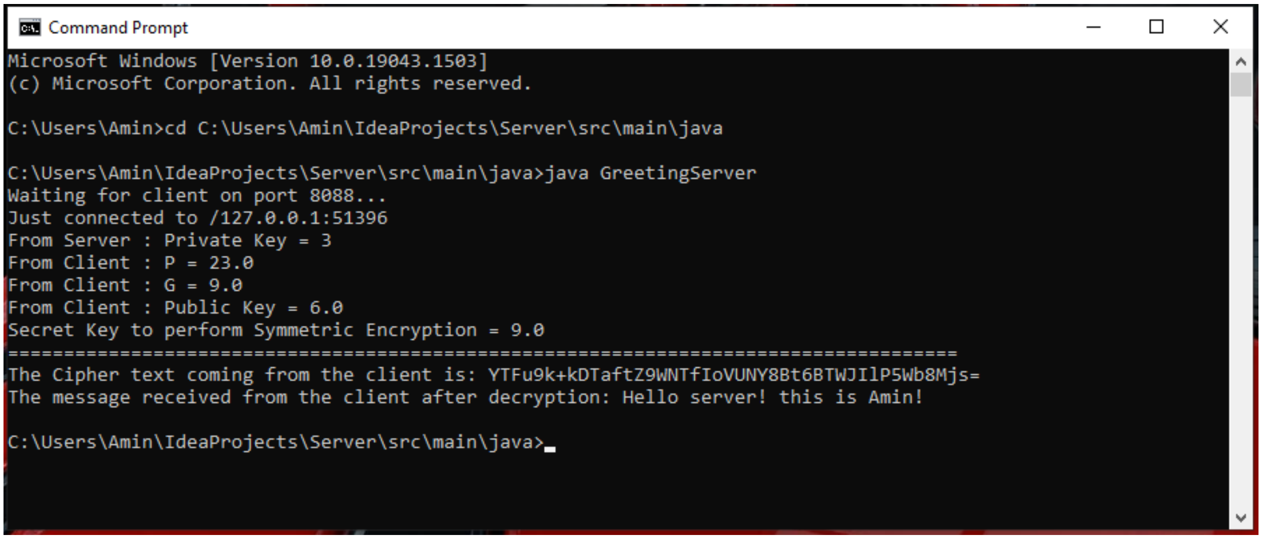
Amin Hassairi

*Now the Client has established a connection with the server and we can see the results from the CMD of the Client:



```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Client\src

C:\Users\Amin\IdeaProjects\Client\src>java GreetingClient
Connecting to localhost on port 8088
Just connected to localhost/127.0.0.1:8088
From Client : Private Key = 4
From Server : Public Key = 16.0
Secret Key to perform Symmetric Encryption = 9.0
The Cipher text to be send to the server is: YTFu9k+kDTaftZ9WNTfIoVUNY8Bt6BTWJIlP5Wb8Mjs=
-------------------------------------------------------
The Cipher text received from the server is: O6t3PlAaXkwcjDDGMLsfbw57Lb9WF0ZpW/pKjgIlra4=
The message received from the server after decryption: Welcome to the server!

C:\Users\Amin\IdeaProjects\Client\src>
```

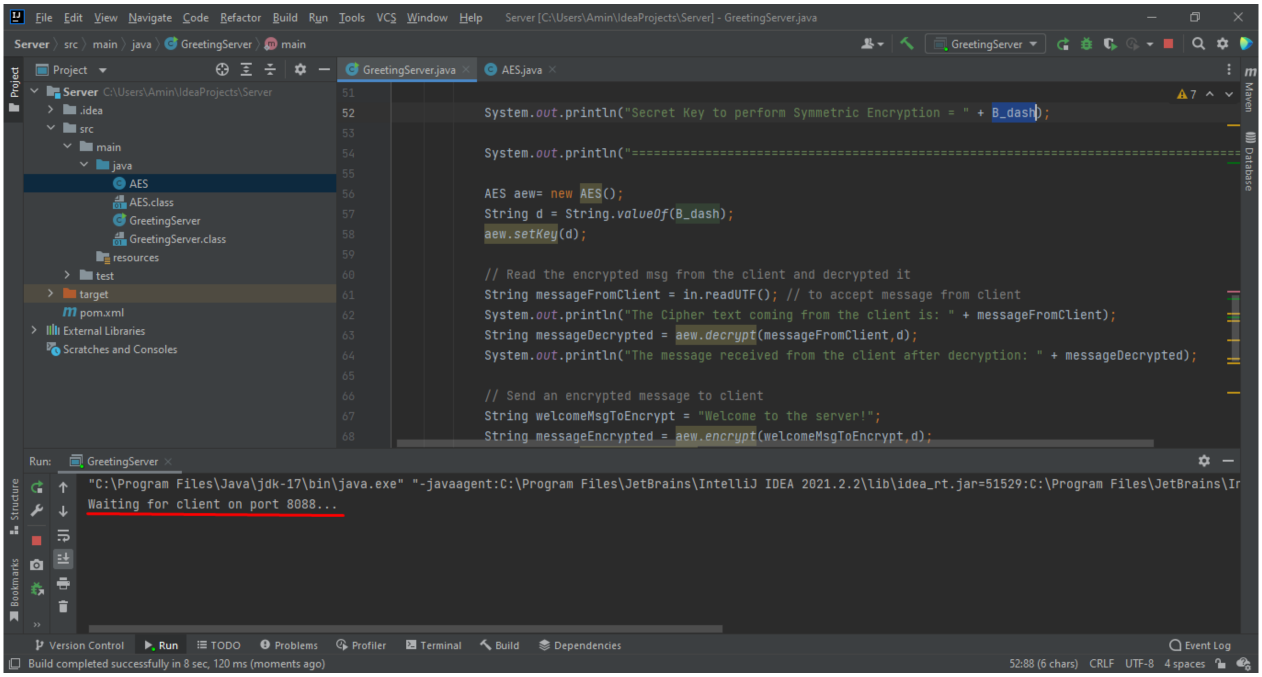
*Also from CMD of the Server we can see the following result:



```
Microsoft Windows [Version 10.0.19043.1503]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amin>cd C:\Users\Amin\IdeaProjects\Server\src\main\java

C:\Users\Amin\IdeaProjects\Server\src\main\java>java GreetingServer
Waiting for client on port 8088...
Just connected to /127.0.0.1:51396
From Server : Private Key = 3
From Client : P = 23.0
From Client : G = 9.0
From Client : Public Key = 6.0
Secret Key to perform Symmetric Encryption = 9.0
=====================================================================================
The Cipher text coming from the client is: YTFu9k+kDTaftZ9WNTfIoVUNY8Bt6BTWJIlP5Wb8Mjs=
The message received from the client after decryption: Hello server! this is Amin!

C:\Users\Amin\IdeaProjects\Server\src\main\java>
```

The same outcome using IntelliJ IDEA:

*Open IntelliJ IDEA and open Server folder. Running GreetingServer.java we can see the following:

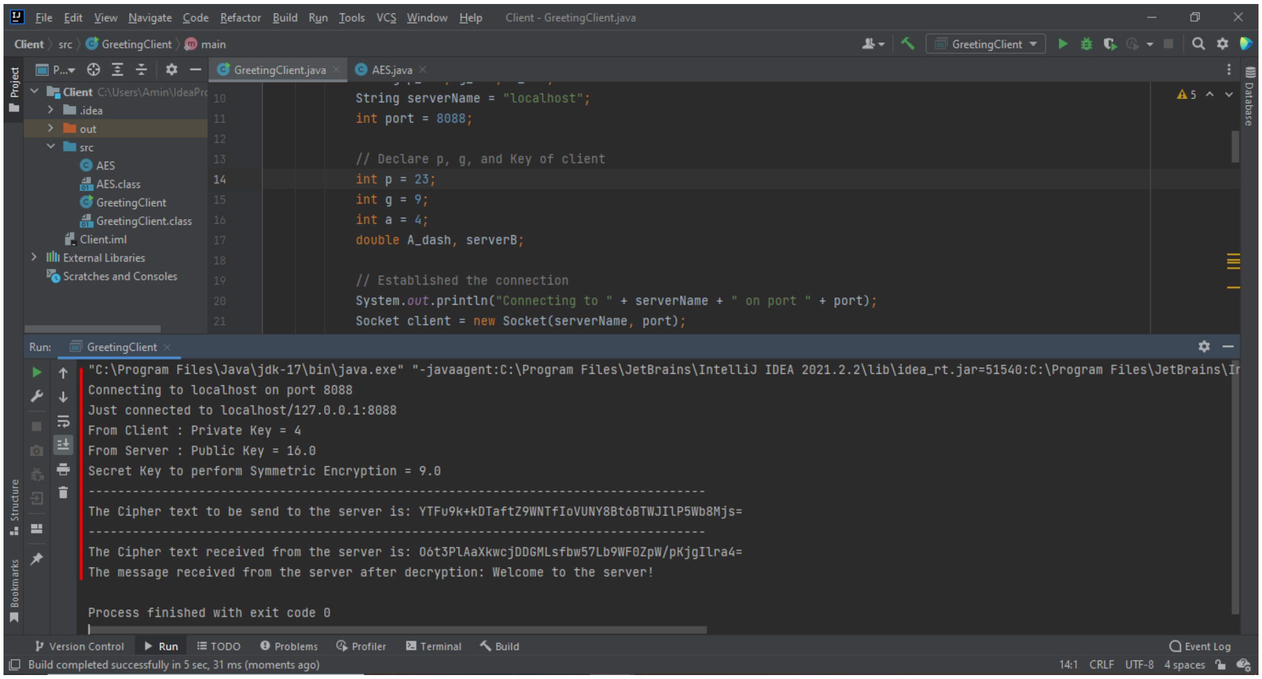


Server [C:\Users\Amin\IdeaProjects\Server] - GreetingServer.java
Server 〉 src 〉 main 〉 java 〉 GreetingServer 〉 main
GreetingServer.java
AES.java
Server C:\Users\Amin\IdeaProjects\Server
.idea
src
main
java
AES
AES.class
GreetingServer
GreetingServer.class
resources
test
target
pom.xml
External Libraries
Scratches and Consoles
Run: GreetingServer
Build completed successfully in 8 sec, 120 ms (moments ago)
52:88 (6 chars) CRLF UTF-8 4 spaces

```java
        System.out.println("Secret Key to perform Symmetric Encryption = " + B_dash);

        System.out.println("==========================================================================

        AES aew= new AES();
        String d = String.valueOf(B_dash);
        aew.setKey(d);

        // Read the encrypted msg from the client and decrypted it
        String messageFromClient = in.readUTF(); // to accept message from client
        System.out.println("The Cipher text coming from the client is: " + messageFromClient);
        String messageDecrypted = aew.decrypt(messageFromClient,d);
        System.out.println("The message received from the client after decryption: " + messageDecrypted);

        // Send an encrypted message to client
        String welcomeMsgToEncrypt = "Welcome to the server!";
        String messageEncrypted = aew.encrypt(welcomeMsgToEncrypt,d);
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.2.2\lib\idea_rt.jar=51529:C:\Program Files\JetBrains\Ir
Waiting for client on port 8088...
```

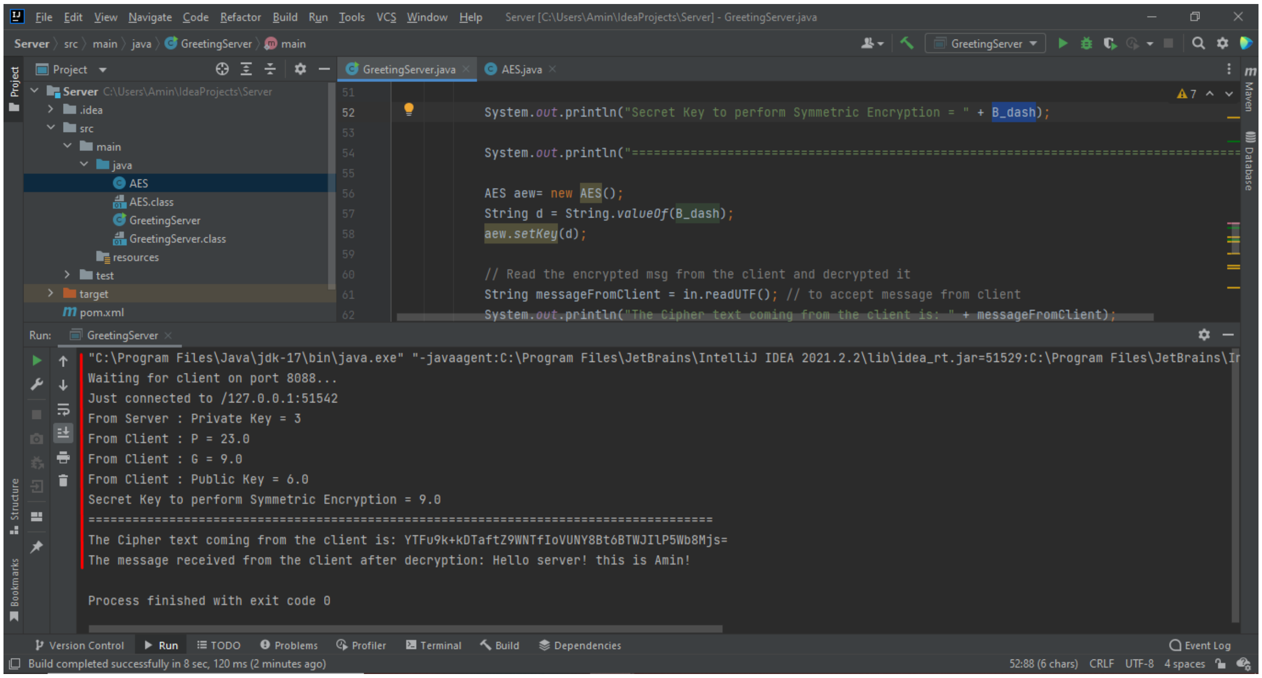
*Open Another window of IntelliJ IDEA and open the Client folder then run GreetingClient.java. We can observe this result:



```java
        String serverName = "localhost";
        int port = 8088;

        // Declare p, g, and Key of client
        int p = 23;
        int g = 9;
        int a = 4;
        double A_dash, serverB;

        // Established the connection
        System.out.println("Connecting to " + serverName + " on port " + port);
        Socket client = new Socket(serverName, port);
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.2.2\lib\idea_rt.jar=51540:C:\Program Files\JetBrains\Ir
Connecting to localhost on port 8088
Just connected to localhost/127.0.0.1:8088
From Client : Private Key = 4
From Server : Public Key = 16.0
Secret Key to perform Symmetric Encryption = 9.0
------------------------------------------------------------------------------------
The Cipher text to be send to the server is: YTFu9k+kDTaftZ9WNTfIoVUNY8Bt6BTWJIlP5Wb8Mjs=
------------------------------------------------------------------------------------
The Cipher text received from the server is: O6t3PlAaXkwcjDDGMLsfbw57Lb9WF0ZpW/pKjgIlra4=
The message received from the server after decryption: Welcome to the server!

Process finished with exit code 0
```

*Switch to the other window of the Server and you can observe the result: Client is connected with the Server.


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Server [C:\Users\Amin\IdeaProjects\Server] - GreetingServer.java
Server 〉 src 〉 main 〉 java 〉 GreetingServer 〉 main
GreetingServer
Project
Server C:\Users\Amin\IdeaProjects\Server
.idea
src
main
java
AES
AES.class
GreetingServer
GreetingServer.class
resources
test
target
pom.xml
GreetingServer.java
AES.java
Run: GreetingServer
Version Control
Run
TODO
Problems
Profiler
Terminal
Build
Dependencies
Event Log
Build completed successfully in 8 sec, 120 ms (2 minutes ago)
52:88 (6 chars) CRLF UTF-8 4 spaces

```java
        System.out.println("Secret Key to perform Symmetric Encryption = " + B_dash);

        System.out.println("==========================================================================

        AES aew= new AES();
        String d = String.valueOf(B_dash);
        aew.setKey(d);

        // Read the encrypted msg from the client and decrypted it
        String messageFromClient = in.readUTF(); // to accept message from client
        System.out.println("The Cipher text coming from the client is: " + messageFromClient);
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.2.2\lib\idea_rt.jar=51529:C:\Program Files\JetBrains\Ir
Waiting for client on port 8088...
Just connected to /127.0.0.1:51542
From Server : Private Key = 3
From Client : P = 23.0
From Client : G = 9.0
From Client : Public Key = 6.0
Secret Key to perform Symmetric Encryption = 9.0
=====================================================================================
The Cipher text coming from the client is: YTFu9k+kDTaftZ9WNTfIoVUNY8Bt6BTWJIlP5Wb8Mjs=
The message received from the client after decryption: Hello server! this is Amin!

Process finished with exit code 0
```