Practical no:- 16

Program Code:

1)

1. Client.java

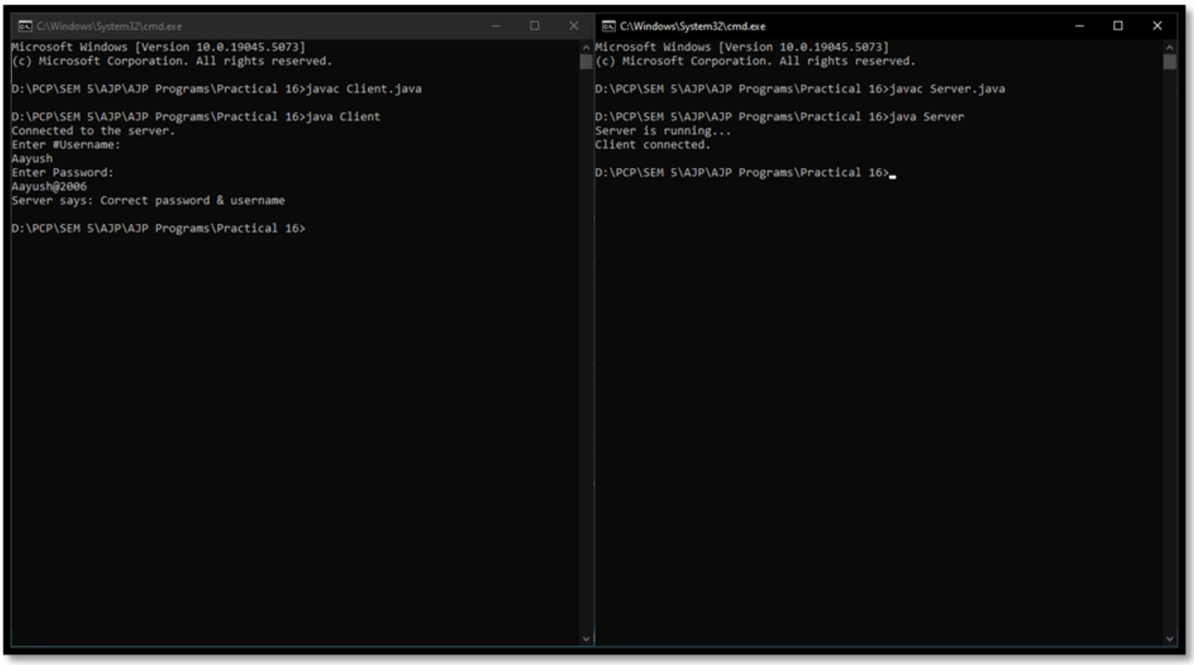
```
import java.net.*;

import java.io.*;


class Client {

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

        Socket clientSocket = new Socket("localhost", 100);

        System.out.println("Connected to the server.");


        BufferedReader userInputReader = new BufferedReader(new
InputStreamReader(System.in));

        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

        BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));


        System.out.println("Enter #Username:");

        String username = userInputReader.readLine();

        out.println(username);


        System.out.println("Enter Password:");

        String password = userInputReader.readLine();

        out.println(password);
```

```
    String serverResponse = in.readLine();

    System.out.println("Server says: " + serverResponse);


    clientSocket.close();

    }

}
```

2. Server.java

```
import java.net.*;

import java.io.*;


class Server {

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

    ServerSocket serverSocket = new ServerSocket(100);

    System.out.println("Server is running...");

    Socket clientSocket = serverSocket.accept();

    System.out.println("Client connected.");


    BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

    PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

    String username = in.readLine();

    String password = in.readLine();

    String validUsername = "Aayush";

    String validPassword = "Aayush@2006";
```

```
          String message;

          if (validUsername.equals(username) && validPassword.equals(password)) {

          message = "Correct password & username";

          } else {

          message = "Incorrect password or username";

          }

          out.println(message);

          clientSocket.close();

          serverSocket.close();

          }

}
```



```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>javac Client.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>java Client
Connected to the server.
Enter #Username:
Aayush
Enter Password:
Aayush@2006
Server says: Correct password & username

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>
```

```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>javac Server.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>java Server
Server is running...
Client connected.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>_
```

Exercise:

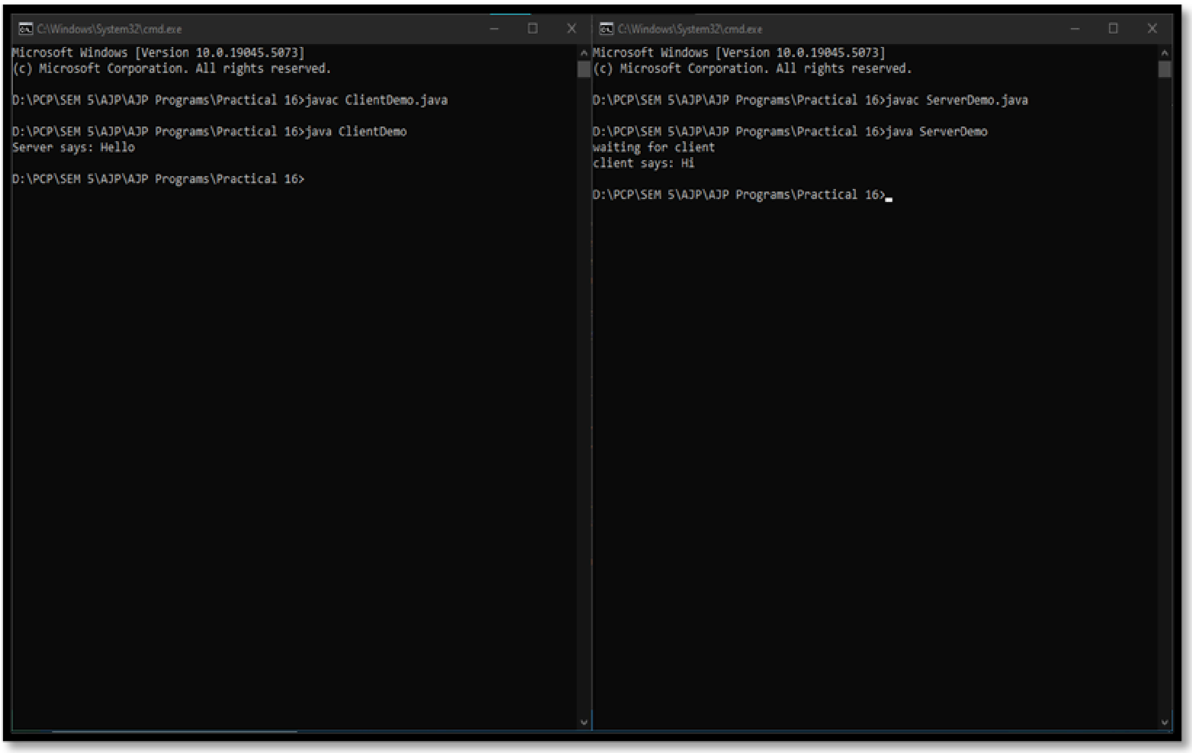
1)

1. ClientDemo.java

```
import java.net.*;
import java.io.*;
class ClientDemo {
        public static void main(String args[]) throws Exception {
        Socket c = new Socket("localhost", 100);
        PrintWriter pw = new PrintWriter(c.getOutputStream(), true);
        pw.println("Hi");
        BufferedReader br = new BufferedReader(new
InputStreamReader(c.getInputStream()));
        String response = br.readLine();
        System.out.println("Server says: " + response);
        br.close();
        pw.close();
        c.close();
        }
}
```

2. ServerDemo.java

```
import java.net.*;
import java.io.*;


class ServerDemo {
        public static void main(String args[]) throws Exception {
        ServerSocket s = new ServerSocket(100);
        System.out.println("waiting for client");
        Socket cs = s.accept();
        BufferedReader br = new BufferedReader(new
InputStreamReader(cs.getInputStream()));

        String str = br.readLine();

        System.out.println("client says: " + str);


        PrintWriter pw = new PrintWriter(cs.getOutputStream(), true);

        pw.println("Hello");

        br.close();

        pw.close();

        cs.close();

        s.close();

        }

}
```



```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>javac ClientDemo.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>java ClientDemo
Server says: Hello

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>
```

```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>javac ServerDemo.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>java ServerDemo
waiting for client
client says: Hi

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>
```

2)

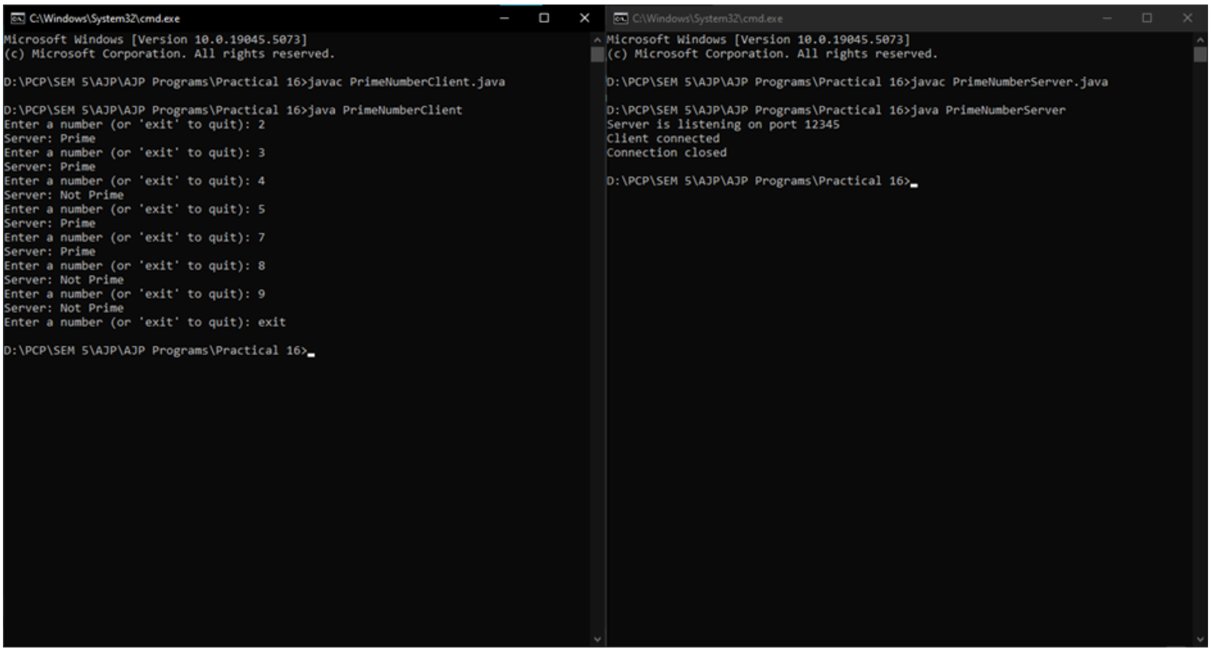
1. PrimeNumberClient.java

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class PrimeNumberClient {
    public static void main(String[] args) {
    try (Socket socket = new Socket("localhost", 12345)) {
    PrintWriter writer = new PrintWriter(socket.getOutputStream(), true);
    BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
    Scanner scanner = new Scanner(System.in);
    String numberStr;
    while (true) {
        System.out.print("Enter a number (or 'exit' to quit): ");
        numberStr = scanner.nextLine();
        if (numberStr.equalsIgnoreCase("exit"))
        break;
        writer.println(numberStr);
        String response = reader.readLine();
      System.out.println("Server: " + response);
    }
    socket.close();
    scanner.close();
    } catch (IOException ex) {
```

```
    ex.printStackTrace();
    }
    }
}
```

2. PrimeNumberServer.java

```
import java.io.*;
import java.net.*;

public class PrimeNumberServer {
    public static void main(String[] args) {
    try (ServerSocket serverSocket = new ServerSocket(12345)) {
    System.out.println("Server is listening on port 12345");
    Socket socket = serverSocket.accept();
    System.out.println("Client connected");

    BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
    PrintWriter writer = new PrintWriter(socket.getOutputStream(), true);

    String numberStr;
    while ((numberStr = reader.readLine()) != null) {
        int number = Integer.parseInt(numberStr);
        String response = isPrime(number) ? "Prime" : "Not Prime";
        writer.println(response);
        socket.close();
    System.out.println("Connection closed");

    } catch (IOException ex) {

    ex.printStackTrace();

    }

    }


    private static boolean isPrime(int number) {

    if (number <= 1)

    return false;

    for (int i = 2; i <= Math.sqrt(number); i++) {

    if (number % i == 0)

        return false;

    }

    return true;

    }

}
```



```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>javac PrimeNumberClient.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>java PrimeNumberClient
Enter a number (or 'exit' to quit): 2
Server: Prime
Enter a number (or 'exit' to quit): 3
Server: Prime
Enter a number (or 'exit' to quit): 4
Server: Not Prime
Enter a number (or 'exit' to quit): 5
Server: Prime
Enter a number (or 'exit' to quit): 7
Server: Prime
Enter a number (or 'exit' to quit): 8
Server: Not Prime
Enter a number (or 'exit' to quit): 9
Server: Not Prime
Enter a number (or 'exit' to quit): exit

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>
```

```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>javac PrimeNumberServer.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>java PrimeNumberServer
Server is listening on port 12345
Client connected
Connection closed

D:\PCP\SEM 5\AJP\AJP Programs\Practical 16>
```