Client.java

```
import java.net.*;

public class DNSClient {
    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {
            // Create a DatagramSocket to send DNS queries to the server on port 9876
            socket = new DatagramSocket();

            // Get user input for the DNS query
            System.out.print("Enter DNS Query: ");
            String dnsQuery = new java.util.Scanner(System.in).nextLine();

            // Send the DNS query to the server
            byte[] sendData = dnsQuery.getBytes();
            InetAddress serverAddress = InetAddress.getByName("localhost"); // Change
this to the server's IP address
            int serverPort = 9876;
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, serverPort);
            socket.send(sendPacket);

            // Receive the DNS response from the server
            byte[] receiveData = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            socket.receive(receivePacket);

            // Display the DNS response
            String ipAddress = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Resolved IP Address: " + ipAddress);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (socket != null && !socket.isClosed()) {
                socket.close();
            }
        }
    }
}
```

Server.java

```
import java.net.*;

public class DNSServer {
    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {
            // Create a DatagramSocket to listen for DNS requests on port 9876
            socket = new DatagramSocket(9876);

            while (true) {
                // Create a DatagramPacket to receive the DNS query
                byte[] receiveData = new byte[1024];
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                socket.receive(receivePacket);

                // Get the DNS query from the received packet
                String dnsQuery = new String(receivePacket.getData(), 0,
receivePacket.getLength());
                System.out.println("Received DNS Query: " + dnsQuery);

                // Perform DNS resolution (you can replace this with actual DNS resolution
logic)
                String ipAddress = resolveDNS(dnsQuery);

                // Send the DNS response back to the client
                byte[] sendData = ipAddress.getBytes();
                InetAddress clientAddress = receivePacket.getAddress();
                int clientPort = receivePacket.getPort();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
                socket.send(sendPacket);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (socket != null && !socket.isClosed()) {
                socket.close();
            }
        }
    }

    private static String resolveDNS(String dnsQuery) {
        // Replace this with your DNS resolution logic
        // For simplicity, this example returns a hardcoded IP address for a specific
domain
        if (dnsQuery.equals("example.com")) {
            return "192.168.1.1";
        } else {
            return "DNS resolution not implemented for this domain";
        }
    }
}
```

Server O/P

```
Received DNS Query: example.com
```

Client O/P

```
Enter DNS Query: example.com
Resolved IP Address: 192.168.1.1
```