A

LAB REPORT

ON

NETWORK PROGRAMMING

By

Prajwal Dahal

TU REG no: 6-2-388-03-2019

Submitted to:

Harendra Subedi

Lecturer

Kantipur College of Management and Information Technology

In partial fulfillment of the requirements for the Course

Network Programming

Mid Baneshwor, Kathmandu

December 2023

TABLE OF CONTENTS

1 Write a program to display IP address of 'google.com' by using InetAddress class.

1.1 Source Code

```
package qstn;
import java.net.InetAddress;
import java.net.UnknownHostException;
public class GoogleIpAddress {
        public static void main(String[] args) {
            try
            {
                   InetAddress address =
                  InetAddress.getByName("www.google.com");
                  System.out.println("IP address: " +
                  address.getHostAddress());
            } catch (UnknownHostException e) {
                System.out.println("Hostname not found");
            }
        }
    }
```

1.2 Output Window

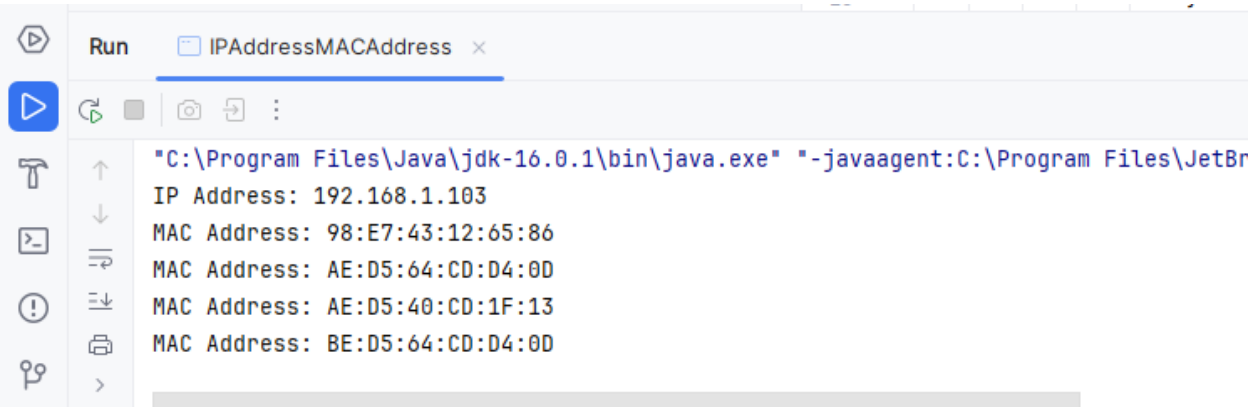
2 Write a program to display mac address and IP Address of your pc.

2.1 Source Code

```
package qstn;
import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.net.UnknownHostException;
import java.util.Arrays;
import java.util.Enumeration;
public class IPAddressMACAddress {
    public static void main(String[] args) throws SocketException,
UnknownHostException {
        InetAddress localhost = InetAddress.getLocalHost();
        String ipAddress = localhost.getHostAddress();
        System.out.println("IP Address: " + ipAddress);
       Enumeration<NetworkInterface> interfaces =
      NetworkInterface.getNetworkInterfaces();
        while (interfaces.hasMoreElements()) {
            NetworkInterface interface1 =
            interfaces.nextElement();
            byte[] macAddress = interface1.getHardwareAddress();
            if (macAddress != null) {
                String macAddressStr = "";
                for (int i = 0; i < macAddress.length; i++) {
                 macAddressStr +=
                 String.format("%02X",macAddress[i]);
                    if (i < macAddress.length - 1) {
```

```
                    macAddressStr += ":";
                }
            }
            System.out.println("MAC Address: " +
            macAddressStr);
        }
    }
}
```

2.2 Output Window



3 Write a program to retrieve protocol, port, host and file from URL.

3.1 Source Code

```
import java.net.URL;
import java.net.MalformedURLException;
public class URLExample {
    public static void main(String[] args) {
        try {
            URL url = new
           URL("https://www.kcmit.edu.np:8080/index.html");
            System.out.println("Protocol: " + url.getProtocol());
            System.out.println("Host: " + url.getHost());
            System.out.println("Port: " + url.getPort());
            System.out.println("File: " + url.getFile());
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
    }
}
```

3.2 Output Window

4 Write a program to construct URI with schema, authority, path, query and fragments.

4.1 Source Code

```
import java.net.URI;
import java.net.URISyntaxException;
public class URITestCreate1 {
    public static void main(String[] args) {
        try {
             URI uri = new URI("https", "kcmit.edu.np", "/path",
            "query=param","DCfragment");
            System.out.println("URI: " + uri.toString());
        } catch (URISyntaxException e) {
            e.printStackTrace();
        }
    }
}
```

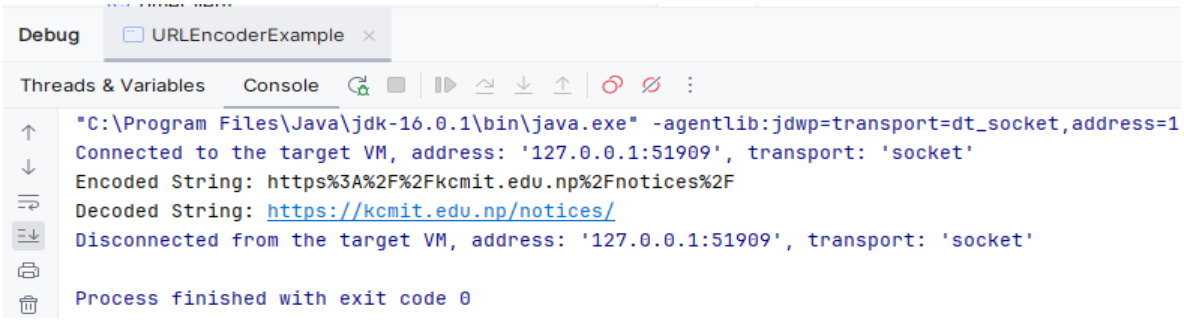
4.2 Output Windows

5 Write a program to demonstrate use of URLEncoder and URLDecoder class.

5.1 Source Code

```
import java.io.UnsupportedEncodingException;
import java.net.URLDecoder;
import java.net.URLEncoder;
public class URLEncoderExample {
    public static void main(String[] args) {
        try {
            String url = "https://kcmit.edu.np/notices/";
             String encodedString = URLEncoder.encode(url, "UTF-
            8");
            System.out.println("Encoded String: " +
            encodedString);
            String decodedString =
            URLDecoder.decode(encodedString, "UTF-8");
            System.out.println("Decoded String: " +
            decodedString);
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
```

5.2 Output Window



6 Write a program to set the cookie and retrieved cookie using CookieManager class.

6.1 Source Code

```
package qstn;
import java.net.*;
import java.util.List;
public class CookieStoreExample {
    public static void main(String[] args) throws
    URISyntaxException {
        CookieManager cookieManager = new CookieManager();
        CookieStore cookieStore = cookieManager.getCookieStore();
        URI uri= new URI("http://example.com");

        HttpCookie cookie = new HttpCookie("name", "value");
        cookie.setDomain(uri.toString());
        cookie.setPath("/");
        cookie.setMaxAge(1000);
        cookieStore.add(uri, cookie);
        List<HttpCookie> retrievedCookie = cookieStore.get(uri);
        System.out.println(retrievedCookie);
        for(HttpCookie cookie1:retrievedCookie){
            System.out.println(cookie1.getName());
            System.out.println(cookie1.getValue());
            System.out.println(cookie1.getDomain());
        }
        cookieStore.remove(uri, cookie);
    }
}
```

6.2 Output Window

7 Write a program to read data from the server.

7.1 Source Code

```
package qstn;
import java.io.*;
import java.net.*;
public class URLDataRetrievalExample {
    public static void main(String[] args)  {
        String url = "https://www.kcmit.com";
        try {
            URL Url = new URL(url);
             HttpURLConnection httpURLConnection = (HttpURLConnection)
           Url.openConnection();
           InputStream inputStream=httpURLConnection.getInputStream();
            InputStreamReader inputStreamReader=new
           InputStreamReader(inputStream);
            BufferedReader reader=new
            BufferedReader(inputStreamReader);
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

7.2 Output Window

8 Write a Program to retrieve specific header fields.

8.1 Source Code

```
package qstn;
import java.io.*;
import java.net.*;
import java.util.*;
public class HeaderFieldExample {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://www.kcmit.edu.np");
            HttpURLConnection connection = (HttpURLConnection)
            url.openConnection();
            String contentType = connection.getContentType();
            int contentLength = connection.getContentLength();
            String contentEncoding =
            connection.getContentEncoding();
            Date date = new Date(connection.getDate());
            Date lastModified = new
            Date(connection.getLastModified());
            Date expires = new Date(connection.getExpiration());
            System.out.println("Content-Type: " + contentType);
            System.out.println("Content-Length: " +
            contentLength);
            System.out.println("Content-Encoding: " +
            contentEncoding);
            System.out.println("Date: " + date);
            System.out.println("Last-Modified: " + lastModified);
            System.out.println("Expires: " + expires);
            connection.disconnect();
```

```
            } catch (IOException e) {
                    e.printStackTrace();
            }
        }
    }
```

8.2 Output Window

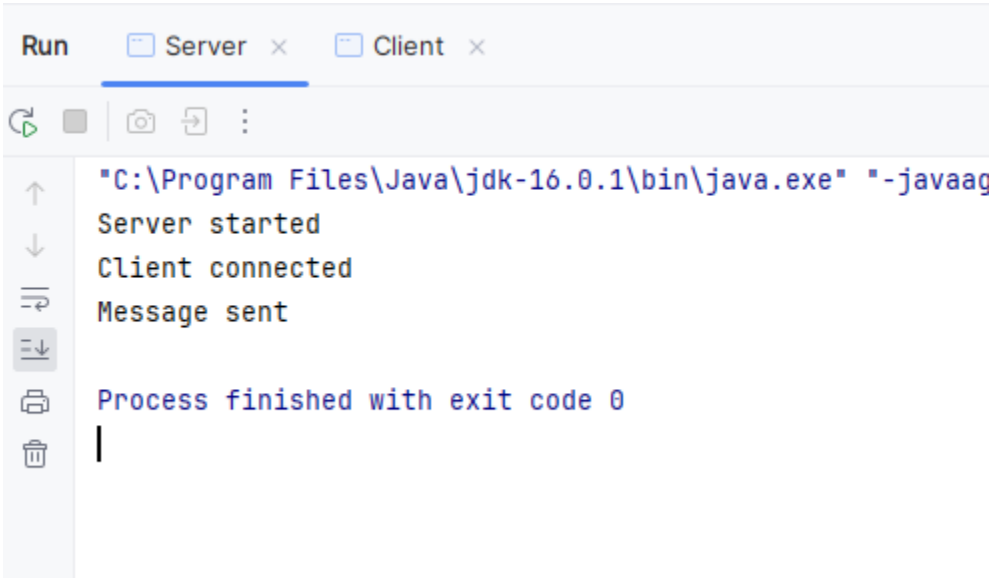
9 Write a program to demonstrate client-server communication using Socket.

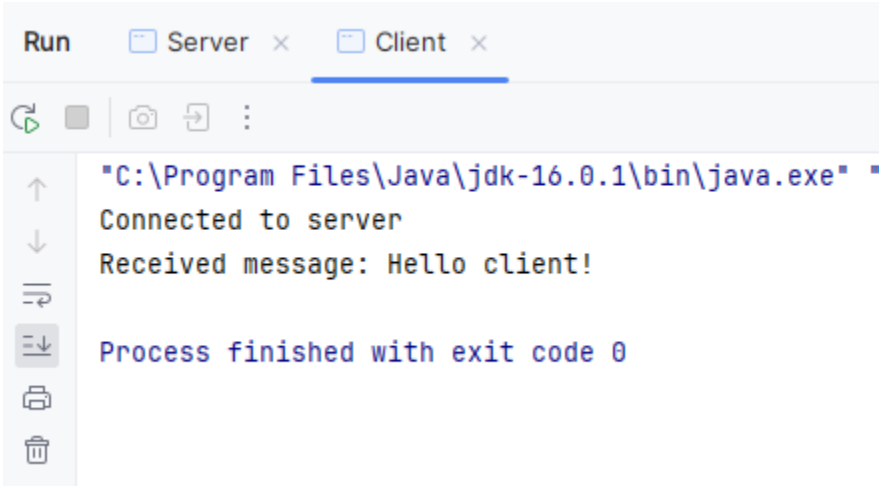
9.1 Source Code

```
package qstn;
import java.io.IOException;
import java.io.OutputStream;
import java.net.ServerSocket;
import java.net.Socket;
public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(12345);
            System.out.println("Server started");
            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected");
            OutputStream outputStream =
           clientSocket.getOutputStream();
            String message = "Hello client!";
            outputStream.write(message.getBytes());
            System.out.println("Message sent");
            serverSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
package qstn;

import java.io.IOException;
import java.io.InputStream;
```

```
import java.net.Socket;

public class Client {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 12345);
            System.out.println("Connected to server");
            InputStream inputStream = socket.getInputStream();
            byte[] buffer= new byte[1234];
            int bytesRead = inputStream.read(buffer);
            String message = new String(buffer, 0, bytesRead);
            System.out.println("Received message: " + message);
            socket.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

9.2 Output Window





10 Write a Program to demonstrate Multithreaded Server.

10.1 Source Code

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;
public class MultiThreadedServer {
    public static void main(String[] args) {
        final int PORT = 12345;
        try (ServerSocket serverSocket = new ServerSocket(PORT)) {
            System.out.println("Server listening on port " + PORT);
            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("New connection from " +
                clientSocket.getInetAddress());
                Thread clientThread = new Thread(() ->
                handleClient(clientSocket));
                clientThread.start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
     private static void handleClient(Socket clientSocket) {
        try (
                Scanner input = new
                Scanner(clientSocket.getInputStream());
```

```
                PrintWriter output = new
                PrintWriter(clientSocket.getOutputStream(), true)
        ) {
            while (input.hasNextLine()) {
                String clientMessage = input.nextLine();
                System.out.println("Received from " +
                clientSocket.getInetAddress() + ": " + clientMessage);
                output.println("Server: " + clientMessage);
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
                 System.out.println("Connection with " +
                clientSocket.getInetAddress() + " closed.");
            try {
                clientSocket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        final String SERVER_HOST = "localhost";
        final int SERVER_PORT = 12345;
```

```
    try (
            Socket socket = new Socket(SERVER_HOST, SERVER_PORT);
            Scanner scanner = new Scanner(System.in);
            PrintWriter output = new
            PrintWriter(socket.getOutputStream(), true)
    ) {
        System.out.println("Connected to the server.");

        Thread receiveThread = new Thread(() -> {
            try (Scanner input = new
            Scanner(socket.getInputStream())) {
                while (input.hasNextLine()) {

                  System.out.println("Server: " +
                  input.nextLine());
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
        receiveThread.start();
        while (true) {
            System.out.print("Client: ");
            String message = scanner.nextLine();
            output.println(message);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
}
```

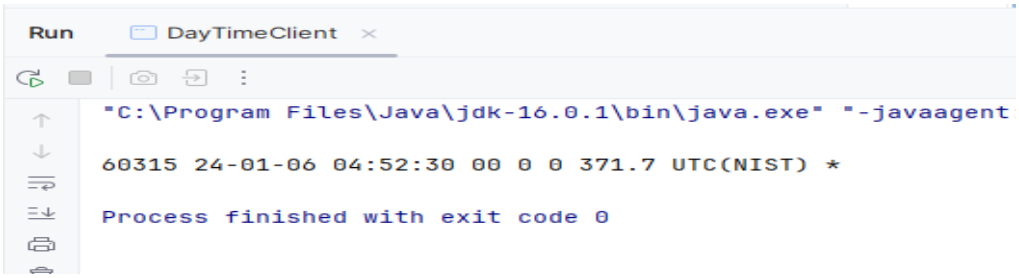
10.2 Output Window

11 Write a program to demonstrate daytime client.

11.1 Source Code

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


public class DayTimeClient {
    public static void main(String[] args) {
        try {
            Socket theSocket = new Socket("time.nist.gov", 13); //
Connect to the server on port 13 (Daytime)
            InputStream timeStream = theSocket.getInputStream();
            BufferedReader bufferedReader=new BufferedReader(new
           InputStreamReader(timeStream));
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
            theSocket.close();
        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

11.2 Output Window



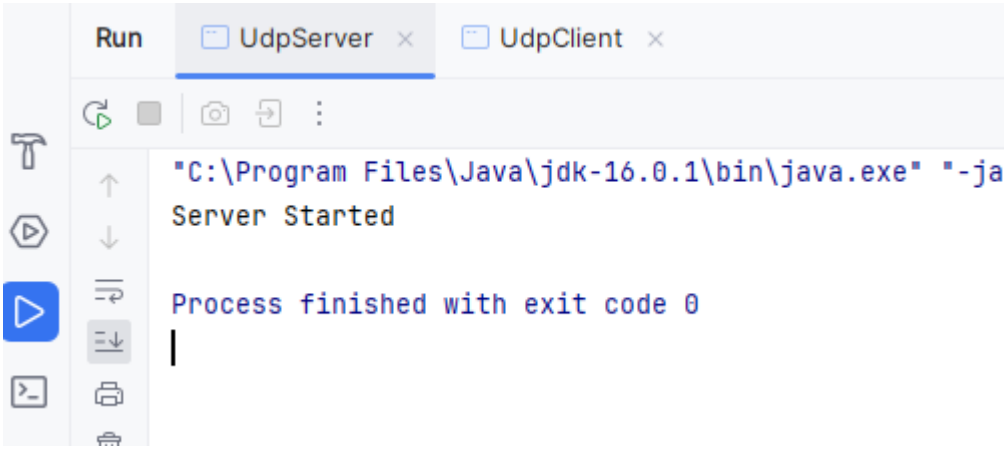
12 Write a program to demonstrate client-server communication using UDP protocol.

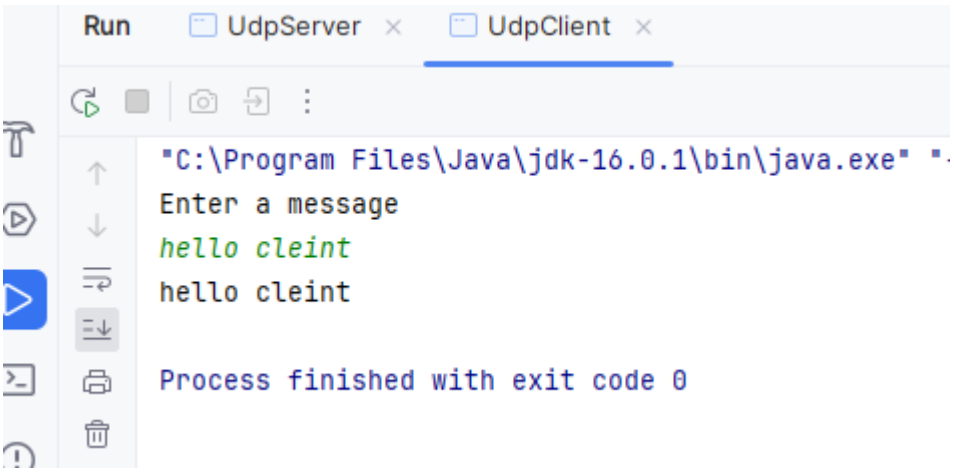
12.1 Source Code

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class UdpServer {
    public static void main(String[] args) throws IOException {
        DatagramSocket datagramSocket=new DatagramSocket(9876);
        byte[]buffer =new byte[1024];
        DatagramPacket datagramPacket=new
        DatagramPacket(buffer,buffer.length);
        datagramSocket.receive(datagramPacket);
        byte[]sendData=datagramPacket.getData();
       DatagramPacket datagramPacket1=new
      DatagramPacket(sendData,sendData.length,datagramPacket.getA
      ddress(),datagramPacket.getPort());
     datagramSocket.send(datagramPacket1);
    }
}

import java.io.IOException;
import java.net.*;
import java.util.Scanner;
public class UdpClient {
    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);
        String message =sc.nextLine();
```

```
        DatagramSocket clientSocket=new DatagramSocket();
        byte[] sendbytes=message.getBytes();
        byte[]buffer=new byte[1024];
        DatagramPacket datagramPacket=new
        DatagramPacket(sendbytes,sendbytes.length,
        InetAddress.getByName("localhost"),9876);
        clientSocket.send(datagramPacket);
        DatagramPacket recieve=new
        DatagramPacket(buffer,buffer.length);
        clientSocket.receive(recieve);
        byte[]x=recieve.getData();
        System.out.println(new String(x).trim());
    }
}
```

12.2 Output Window




Run
UdpServer
UdpClient
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "
Enter a message
hello cleint
hello cleint
Process finished with exit code 0

13 Write a Program to demonstrate multicast communication.

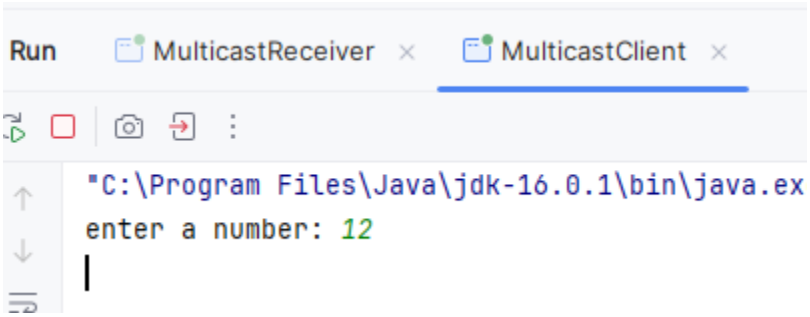
13.1 Source Code

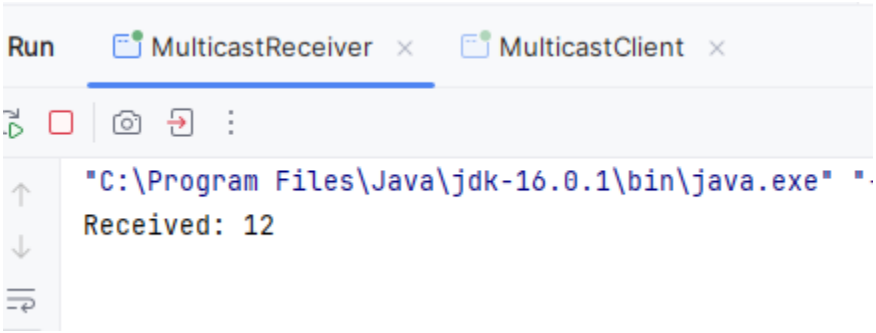
```
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
public class MulticastReceiver {
    public static void main(String[] args) {
        try {
            int port = 12345;
            InetAddress group = InetAddress.getByName("239.0.0.1");
            MulticastSocket socket = new MulticastSocket(port);
            socket.joinGroup(group);
            byte[] buffer = new byte[1024];
            while (true) {
                DatagramPacket packet = new DatagramPacket(buffer,
                 buffer.length);
                socket.receive(packet);
                String message = new String(packet.getData(), 0,
                packet.getLength());
                System.out.println("Received: " + message);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
import java.io.IOException;
import java.net.*;
import java.util.Scanner;
public class MulticastClient {
    public static void main(String[] args) {
        try {
            System.out.print("enter a number: ");
            Scanner sc= new Scanner(System.in);
            String message =sc.nextLine();
            byte[] sendBuffer = message.getBytes();
            InetAddress multicastGroup =
           InetAddress.getByName("239.0.0.1");
            int port = 12345;
            MulticastSocket socket = new MulticastSocket();
            socket.joinGroup(multicastGroup);
            DatagramPacket sendPacket = new DatagramPacket(sendBuffer,
            sendBuffer.length, multicastGroup, port);
            socket.send(sendPacket);
            byte[] receiveBuffer = new byte[1024];
            DatagramPacket receivePacket = new
           DatagramPacket(receiveBuffer, receiveBuffer.length);
            socket.receive(receivePacket);
            String receivedMessage = new
            String(receivePacket.getData(),
            0,receivePacket.getLength());
            System.out.println("Server says: " + receivedMessage);

            socket.leaveGroup(multicastGroup);
            socket.close();
        } catch (IOException e) {
```

```
            e.printStackTrace();
        }
    }
}
```

13.2 Output Window





14 Write a Program to demonstrate secure socket.

14.1 Source Code

```
import javax.net.ssl.HandshakeCompletedEvent;
import javax.net.ssl.HandshakeCompletedListener;
import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;
import java.io.*;
public class HTTPClient {
    public static void main(String[] args) {
        int port = 443;
        String host = "www.usps.com";
        try {
            SSLSocketFactory factory
            = (SSLSocketFactory) SSLSocketFactory.getDefault( );
            SSLSocket socket = (SSLSocket)
            factory.createSocket(host, port);

            socket.addHandshakeCompletedListener(handshakeComplete
            dEvent -> System.out.println("handshake completed"));
            String[] supported = socket.getSupportedCipherSuites(
           );
           socket.setEnabledCipherSuites(supported);
           Writer out = new
      OutputStreamWriter(socket.getOutputStream( ));
           out.write("GET http://" + host + "/ HTTP/1.1\r\n");
           out.write("Host: " + host + "\r\n");
           out.write("\r\n");
            out.flush( );
           BufferedReader in = new BufferedReader(
```

```
                new InputStreamReader(socket.getInputStream( )));
                String s;
                while (!(s = in.readLine( )).equals("")) {
                  System.out.println(s);
                }
                System.out.println( );
               String contentLength = in.readLine( );
                int length = Integer.MAX_VALUE;
                 try {
                   length = Integer.parseInt(contentLength.trim( ), 16);
                 }
                 catch (NumberFormatException ex) {
                       ex.printStackTrace();
                  }
                  System.out.println(contentLength);
                  String c;
                  int i = 0;
                  while ((c = in.readLine() )!=null  && i++ < length) {
                              System.out.println(c);
                          }
                 System.out.println( );
                 out.close( );
                  in.close( );
                   socket.close( );
}catch (IOException ex) {
            System.err.println(ex);
   }
 }}
```

14.2 Output Windows

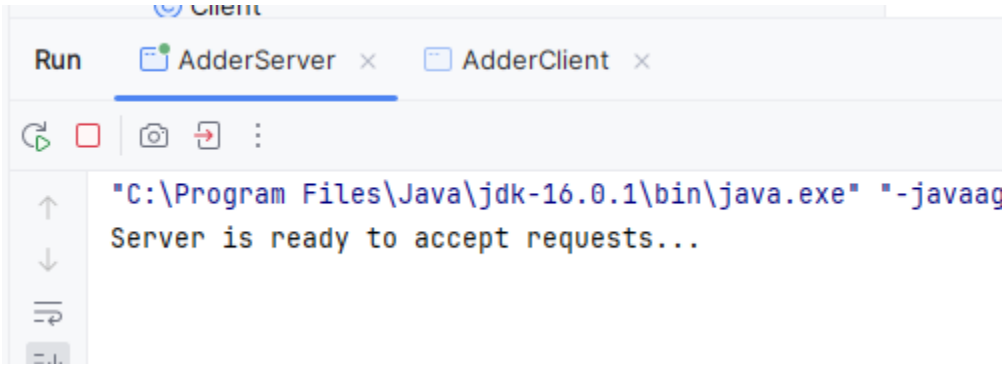
15 Write a program to add two numbers using RMI.

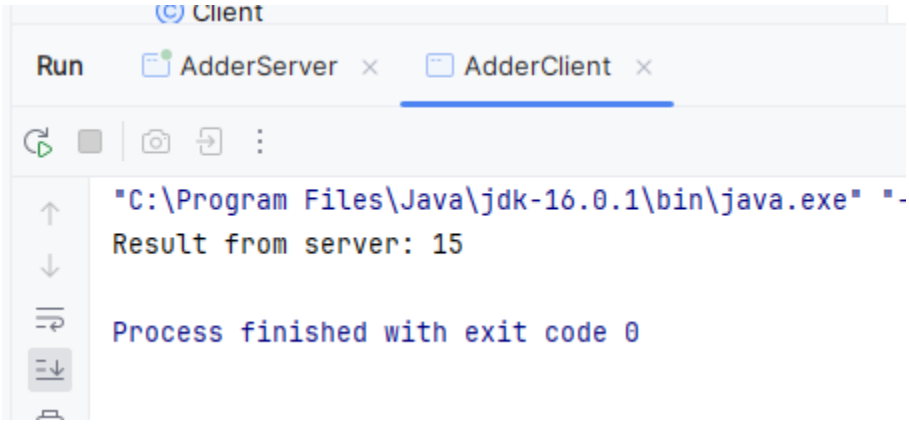
15.1 Source code

```
package RMI;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface Adder extends Remote {
    int add(int a, int b) throws RemoteException;
}
package RMI;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
public class AdderImpl extends UnicastRemoteObject implements Adder {
    public AdderImpl() throws RemoteException {
        super();
    }

    public int add(int a, int b) throws RemoteException {
        return a + b;
    }
}
package RMI;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class AdderServer {
    public static void main(String[] args) {
        try {
            Adder adder = new AdderImpl();
            Registry registry = LocateRegistry.createRegistry(1099);
            registry.rebind("Adder", adder);
```

```
            System.out.println("Server is ready to accept
            requests...");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
package RMI;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class AdderClient {
    public static void main(String[] args) {
        try {
            Registry registry =
LocateRegistry.getRegistry("localhost", 1099);
            Adder adder = (Adder) registry.lookup("Adder");

            int result = adder.add(5, 10);
            System.out.println("Result from server: " + result);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

15.2 Output Window


Client
Run
AdderServer
AdderClient
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "-javaag
Server is ready to accept requests...


Client
Run
AdderServer
AdderClient
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "-
Result from server: 15
Process finished with exit code 0

16 Write program to implement the concept on Filling and Draining buffer, Duplicating buffer, Slicing buffer and Compact buffer.

16.1 Source code

```
package qstn;
import java.nio.ByteBuffer;
public class ByteBufferOperations {
    public static void main(String[] args) {
        ByteBuffer buffer = ByteBuffer.allocate(10);
        for (int i = 1; i <= 5; i++) {
            buffer.put((byte) i);
        }
       System.out.println("Original Buffer: " +
      byteArrayToHexString(buffer.array()));
        buffer.flip();
        while (buffer.hasRemaining()) {
            System.out.println("Drained Byte: " + buffer.get());
        }
        ByteBuffer duplicateBuffer = buffer.duplicate();
         System.out.println("Duplicated Buffer: " +
      byteArrayToHexString(duplicateBuffer.array()));
        buffer.clear();
        buffer.put(new byte[]{1, 2, 3, 4, 5, 6, 7, 8, 9, 10});
        buffer.position(3).limit(7);
        ByteBuffer slicedBuffer = buffer.slice();
         System.out.println("Sliced Buffer: " +
      byteArrayToHexString(slicedBuffer.array()));
        buffer.clear();
        buffer.put(new byte[]{1, 2, 3, 4, 5});
        buffer.flip();
```

```
        buffer.get();
        buffer.compact();
       System.out.println("Compacted Buffer: " +
    byteArrayToHexString(buffer.array()));
    }
    private static String byteArrayToHexString(byte[] array) {
        StringBuilder hexString = new StringBuilder();
        for (byte b : array) {
            hexString.append(String.format("%02X ", b));
        }
        return hexString.toString().trim();
    }
}
```

16.2 Output window