Objective: To learn about the Network Programming in Java Programming Language

Theory:

Network Programming

Network

A network is a system of interconnected elements that share resources, communicate, or collaborate to achieve specific tasks or goals. The term can refer to various fields, but its meaning generally involves a structure where connections exist between different entities.

Network Programming

Network programming refers to writing programs or software that enable devices to communicate with each other over a network, such as the internet or a local area network (LAN). It involves the use of specific protocols, APIs, and libraries to manage the sending, receiving, and processing of data between computers, servers, or other devices

Before dive into network programming language we need to clear about the concept likes :

1. Protocols
2. Sockets
3. Client-Server Model
4. Asynchronous and Multithreading
5. Security

In this lab we do network programming in java . There are various programming language which supports networking such as Python, JavaScript etc.

Advantage of choosing Java as a programming language for networking are :

- Object-Oriented Programming (OOP)
- Rich API and Libraries
- Robust and Secure
- Scalability
- Multithreading and Concurrency
- Active Community and Long-Term Support
- Cross-Platform Development (Mobile, Web, and Desktop)
- Cloud and Microservices
- Platform Independence (Write Once, Run Anywhere)

Lab work :

1. Write a program that prints the address of www.tufohss.edu.np

```
import java.net.*;
public class NameExample {
 public static void main (String[] args) {
 try {
        InetAddress address = InetAddress.getByName("www.tufohss.edu.np");
        System.out.println(address);
 } catch (UnknownHostException ex) {
        System.out.println("Could not find.");
 }
 }
}
```

2. Write a program that finds the address of the local machine

```
import java.net.*;
public class MyAddress {
 public static void main (String[] args) {
 try {
        InetAddress address = InetAddress.getLocalHost();
        System.out.println(address);
} catch (UnknownHostException ex) {
        System.out.println("Could not find this computer's address.");
 }
 }
}
```

3. Write a program that finds the canonical hostname of a given the address

```
 import java.net.*;
public class ReverseTest {
 public static void main (String[] args) throws UnknownHostException {
 InetAddress ia = InetAddress.getByName("182.93.84.136");
 System.out.println(ia.getCanonicalHostName());
 }
}
```

4. Write a program to find the IP address and Host name of the local machine

```
 import java.net.*;
public class MyAddress {
 public static void main(String[] args) {
 try {
        InetAddress me = InetAddress.getLocalHost();
        String dottedQuad = me.getHostAddress();
        System.out.println("My address is " + dottedQuad);
        String hostname = me.getHostName();
        System.out.println("Local host name: "+hostname);
        } catch (UnknownHostException ex) {
        System.out.println("I'm sorry. I don't know my own address."); }
 }
}
```

5. Write a program to get IPv4 and IPv6 address of a given webaddress

```
import java.net.*;
    public class Inet46Address {
     public static void main(String[] a){
     try{
     InetAddress addr = InetAddress.getByName("ipv6.google.com");     if(addr
    instanceof Inet6Address) {
     System.out.println("IPv6 = " + addr.getHostAddress());  }
     if(addr instanceof Inet4Address) {
     System.out.println("IPv4 = " + addr.getHostAddress());  }
     }
     catch(Exception e){
     }
     }
    }
```

6. Write a program for determining whether an IP address is IPv4 or IPv6

```
import java.net.*;
    public class AddressTests {
    public static void main(String[] args) {
     try {
                InetAddress ia = InetAddress.getByName("FF00::");
                byte[] address = ia.getAddress();
                if(address.length == 4)
     System.out.println("IPv4");
                else if(address.length == 16)
     System.out.println("IPv6");
     } catch (UnknownHostException ex) {
                System.out.println("I'm sorry. I don't know my own address."); }
    }
```

7. Write a program for testing the characteristics of an IP address

```
import java.net.*;
    public class IPCharacteristics {
     public static void main(String[] args) {

     try {
     InetAddress address = InetAddress.getByName(args[0]);
     if (address.isAnyLocalAddress()) {
                System.out.println(address + " is a wildcard address.");
     }
     if (address.isLoopbackAddress()) {
                System.out.println(address + " is loopback address.");
     }
    if (address.isLinkLocalAddress()) {
                System.out.println(address + " is a link-local address.");
     } else if (address.isSiteLocalAddress()) {
                System.out.println(address + " is a site-local address.");
     } else {
                System.out.println(address + " is a global address.");
     }
```

```
if (address.isMulticastAddress()) {
if (address.isMCGlobal()) {
        System.out.println(address + " is a global multicast address.");
} else if (address.isMCOrgLocal()) {
        System.out.println(address + " is an organization wide multicast address.");
} else if (address.isMCSiteLocal()) {
        System.out.println(address + " is a site wide multicast address.");
} else if (address.isMCLinkLocal()) {
        System.out.println(address + " is a subnet wide multicast address.");
} else if (address.isMCNodeLocal()) {
        System.out.println(address + " is an interface-local multicast address.");
} else {
        System.out.println(address + " is an unknown multicast address type.");
}
} else {
        System.out.println(address + " is a unicast address.");
}
} catch (UnknownHostException ex) {
        System.err.println("Could not resolve " + args[0]);
}
}
}
```

8. Write a program that compare the domain name are “www.ibiblio.org” and “helios.ibiblio.org” the same?

```
import java.net.*;
public class IBiblioAliases {
 public static void main (String args[]) {
 try {
        InetAddress ibiblio = InetAddress.getByName("www.ibiblio.org");
        InetAddress helios = InetAddress.getByName("helios.ibiblio.org");
 if (ibiblio.equals(helios)) {
        System.out.println ("www.ibiblio.org is the same as helios.ibiblio.org");
 } else {
        System.out.println ("www.ibiblio.org is not the same as helios.ibiblio.org");
 }
 } catch (UnknownHostException ex) {
        System.out.println("Host lookup failed.");
 }
 }
}
```

9. Write a program that lists all the network interfaces.

```
import java.net.*;
import java.util.*;
public class InterfaceLister {
 public static void main(String[] args) throws SocketException {
        Enumeration<NetworkInterface> interfaces = NetworkInterface.getNetworkInterfaces(); while
        (interfaces.hasMoreElements()) {
                NetworkInterface ni = interfaces.nextElement(); // get elements from the enumuration.
                System.out.println(ni);
        }
}
```

```
}
```

10. Write a program that use of network interfaces Getter methods

```
import java.net.*;
import java.util.*;
public class InterfaceLister {
 public static void main(String[] args) throws SocketException {
        NetworkInterface in = NetworkInterface.getByName("lo"); //lo = interface name
Enumeration<InetAddress> enu = in.getInetAddresses();
 while (enu.hasMoreElements()) {
 InetAddress inetAddress = enu.nextElement();
        String hostname = inetAddress.getHostAddress();
 System.out.println("host: "+ hostname);
 }
 }
}
```

11. Write a program to check remote system is reachable or not [Tesing Reachability]

```
import java.net.*;
 public class TestingReachabilityEx {
 public static void main(String[] a){
 try{
 InetAddress net = InetAddress.getByName("22.22.22.22");
if(net.isReachable(100))
 System.out.println("Success");
 else
 System.out.println("Failed");
 }
 catch(Exception e){
 }
 }
}
```

12. Write a program that demonstrate the SpamCheck.

```
import java.net.*;
class SpamCheck {
 public static final String BLACKHOLE = "sbl.spamhaus.org";
 public static void main(String[] args) throws UnknownHostException { try{
                InetAddress address = InetAddress.getByName("202.15.5.111");
                byte[] quad = address.getAddress();
                String query = BLACKHOLE;
                for (byte octet : quad) {
                        int unsignedByte = octet < 0 ? octet + 256 : octet;
                        query = unsignedByte + "." + query;
                }
                //query = 202.5.5.10.sbl.spamhaus.org
                InetAddress.getByName(query);
                System.out.println("No Spam");
        }
        catch(Exception e)
        {
                System.out.println("Spam");
```

```
        }
    }
    private static boolean isSpammer(String arg) {
        try {
            InetAddress address = InetAddress.getByName(arg);
            byte[] quad = address.getAddress();
            String query = BLACKHOLE;
            for (byte octet : quad) {
            int unsignedByte = octet < 0 ? octet + 256 : octet;
                query = unsignedByte + "." + query;
            }
            //query = 202.5.5.10.sbl.spamhaus.org
            InetAddress.getByName(query);
            return true;
         } catch (UnknownHostException e) {
            return false;
          }
        }
    }
```

13. Write a program to process web server logfiles.

```
import java.io.*;
import java.net.*;
public class ServerLogs {
 public static void main(String[] args) {
 try {
 FileInputStream fin = new FileInputStream("access.log");
InputStreamReader in = new InputStreamReader(fin);
 BufferedReader bins = new BufferedReader(in);
 stream  for (String entry = bins.readLine();entry != null;entry = bins.readLine()) {
      int index = entry.indexOf(' ');
String ip = entry.substring(0, index);
      String theRest = entry.substring(index);
System.out.println("Rest Info="+theRest);
try {
InetAddress address = InetAddress.getByName(ip);
System.out.println("IP="+ip);
System.out.println("Host="+address.getHostName());
} catch (UnknownHostException ex) {
System.err.println("Host Is Not Resolved.");
}
}
bins.close();
 } catch (IOException ex) {
     System.out.println("Exception: " + ex);
 }
 }
}
```

Chapter – 3: URLs and URIs

1. Write a program that splits the parts of a URL [Splitting URL into pieces information]

```
import java.net.*;
public class URLSplitter {
 public static void main(String args[]) {
 for (int i = 0; i < args.length; i++) {
        try {
            URL u = new URL(args[i]);
            System.out.println("The URL is " + u);
            System.out.println("The scheme is " + u.getProtocol());
            System.out.println("The user info is " + u.getUserInfo());
            String host = u.getHost();
            System.out.println("The host is " + host);
            System.out.println("The port is " + u.getPort());
            System.out.println("The path is " + u.getPath());
            System.out.println("The ref is " + u.getRef());
            System.out.println("The query string is " + u.getQuery());
        } catch (MalformedURLException ex) {
            System.err.println(args[i] + " is not a URL I understand.");
        }
        System.out.println();
        }
 }
}
```

2. Write a program that checks the which protocols does a virtual machine support or not?

```
import java.net.*;
public class ProtocolTester {
 public static void main(String[] args) {
 // hypertext transfer protocol
 testProtocol("http://www.adc.org");
// secure http
 testProtocol("https://www.amazon.com/exec/obidos/order2/");
 // file transfer protocol
 testProtocol("ftp://ibiblio.org/pub/languages/java/javafaq/");
 // Simple Mail Transfer Protocol
 testProtocol("mailto:elharo@ibiblio.org");
 // telnet
 testProtocol("telnet://dibner.poly.edu/");
 // local file access
 testProtocol("file:///etc/passwd");
 // gopher
 testProtocol("gopher://gopher.anc.org.za/");
}

private static void testProtocol(String url) {
 try {
        URL u = new URL(url);
        System.out.println(u.getProtocol() + " is supported");
 } catch (MalformedURLException ex) {
        String protocol = url.substring(0, url.indexOf(':'));
```

```
                System.out.println(protocol + " is not supported");
            }
            }
           }
```

3. Write a program to download a web page of a given web address.

```
import java.io.*;
import java.net.*;
public class SourceViewer {
   public static void main(String[] args) {
      String urlString = "https://cody.com.np/";
      String outputFileName = "DownloadedPage.html";
      try (BufferedReader reader =new BufferedReader(new
InputStreamReader(new URL(urlString).openStream()));
          BufferedWriter writer = new BufferedWriter(new
FileWriter(outputFileName))) {
          String line;
          while ((line = reader.readLine()) != null) {
             writer.write(line);
             writer.newLine();
          }
          System.out.println("Successfully downloaded the webpage content
to: " + outputFileName);
      } catch (MalformedURLException e) {
          System.out.println("Invalid URL: " + urlString);
      } catch (IOException e) {
          System.out.println("Error reading or writing data: " +
e.getMessage());
      }
   }
}
```

4. Write a program for resolving relatives URI

```
        String uribase = "https://www.test.org/";

        String urirelative = "languages/../java";
        URI uriBase = new URI(uribase);

        // create() method
        URI uri = URI.create(str);
       System.out.println("Base URI = " + uriBase.toString());
       URI uriRelative = new URI(urirelative);
        System.out.println("Relative URI = " + uriRelative.toString());
        URI uriResolved = uriBase.resolve(uriRelative);
        System.out.println("Resolved URI = " + uriResolved.toString())
```

5. Write a program to download an object

```
import java.io.*;
import java.net.*;
public class ContentGetter {
 public static void main (String[] args) {
 if (args.length > 0) {
        // Open the URL for reading
        try {
            URL u = new URL(args[0]);
            Object o = u.getContent();
            System.out.println("I got a " + o.getClass().getName());
        } catch (MalformedURLException ex) {
            System.err.println(args[0] + " is not a parseable URL");
        } catch (IOException ex) {
            System.err.println(ex);
        }
 }
 }
}
```

6. Write a program to demonstrate the x-www-form-urlencoded strings

```
import java.io.*;
import java.net.*;
public class EncoderTest {
 public static void main(String[] args) {
 try {
        //Encoder Example
         System.out.println(URLEncoder.encode("This string has spaces","UTF-8"));
         System.out.println(URLEncoder.encode("This*string*has*asterisks","UTF-8"));
         System.out.println(URLEncoder.encode("This%string%has%percent%signs","UTF-8"));
         System.out.println(URLEncoder.encode("This+string+has+pluses", "UTF-8"));
         System.out.println(URLEncoder.encode("This/string/has/slashes", "UTF-8"));
         System.out.println(URLEncoder.encode("This\"string\"has\"quote\"marks","UTF-8"));
         System.out.println(URLEncoder.encode("This:string:has:colons", "UTF-8"));
         System.out.println(URLEncoder.encode("This~string~has~tildes", "UTF-8"));
         System.out.println(URLEncoder.encode("This(string)has(parentheses)","UTF-8"));
         System.out.println(URLEncoder.encode("This.string.has.periods", "UTF-8"));
         System.out.println(URLEncoder.encode("This=string=has=equals=signs","UTF-8"));
        System.out.println(URLEncoder.encode("This&string&has&ampersands","UTF-8"));
        System.out.println(URLEncoder.encode("Thiséstringéhasé non-ASCII characters", "UTF-8"));
        //Decoder Example
        System.out.println(URLDecoder.decode This%3Astring%3Ahas%3Acolons", "UTF-8")); }
 catch (UnsupportedEncodingException ex) {
        throw new RuntimeException("Broken VM does not support UTF-8");
 }
 }
}
```

7. Write a program that communicating with Server-Side Programs Through GET

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

public class ServerExample {
   public static void main(String[] args) {
      System.out.print("Enter keyword: ");
      Scanner scan = new Scanner(System.in);
      String search = scan.nextLine();
      // Construct the search URL based on the user input
      String urlString = "https://www.bing.com/search?q=" + search;

      try {
         URL url = new URL(urlString);
         // Try-with-resources to automatically close InputStream and Scanner
         try (InputStream in = new BufferedInputStream(url.openStream());
              InputStreamReader theHTML = new InputStreamReader(in)) {
            int c;
            while ((c = theHTML.read()) != -1) {
               System.out.print((char) c);
            }
         } catch (IOException e) {
            System.err.println("Error retrieving data from the URL: " + e.getMessage());
         }
      } catch (MalformedURLException e) {
         System.err.println("Invalid URL format: " + urlString);
      } finally {
         scan.close();
      }
   }
}
```

1. Write a program that shows a simple CookiePolicy that blocks cookies from .gov domains, but allows others.

```
import java.net.*;
public class CookieExample implements CookiePolicy {
 @Override
 public boolean shouldAccept(URI uri, HttpCookie cookie) {
 if (uri.getAuthority().toLowerCase().endsWith(".gov") || cookie.getDomain().toLowerCase().endsWith(".gov"))
{
  return false;
 }
 return true;
 }
}
```

2. Program to implement the CookieStore Methods (add, read, delete) cookies

```
import java.io.*;
import java.net.*;
import java.util.*;
public class CookiesManagerEx{
public static void main(String[] args)
{
CookieManager cookieManager = new CookieManager();
CookieStore cookieStore = cookieManager.getCookieStore();
// creating cookies and URI
HttpCookie cookieA = new HttpCookie("First", "1");
HttpCookie cookieB = new HttpCookie("Second", "2");
URI uri= URI.create("https://www.ambition.edu.np/");
// Method 1 - add(URI uri, HttpCookie cookie)
cookieStore.add(uri, cookieA);
cookieStore.add(null, cookieB);
System.out.println("Cookies successfully added\n");
List cookiesWithURI = cookieStore.get(uri);
System.out.println("Cookies associated with URI in CookieStore : "+ cookiesWithURI + "\n");
List cookieList = cookieStore.getCookies();
System.out.println("Cookies in CookieStore : "+ cookieList + "\n");
// Method 4 - getURIs()
List uriList = cookieStore.getURIs();
System.out.println("URIs in CookieStore" + uriList+ "\n");

System.out.println("Removal of Cookie : "+ cookieStore.remove(uri, cookieA)); List
remainingCookieList = cookieStore.getCookies();
System.out.println("Remaining Cookies : " + cookieList + "\n"); //
Method 6 - removeAll()
System.out.println("Removal of all Cookies : "+ cookieStore.removeAll()); List
EmptyCookieList = cookieStore.getCookies();
System.out.println("Empty CookieStore : " + cookieList);
}
}
```

Chapter – 5 : URL Connections

1. Write a program to download a web page using URLConnection.

```
import java.io.*;
import java.net.*;
public class SourceViewer2 {
public static void main (String[] args) {
try {
        // Open the URLConnection for reading
        URL u = new URL("https://www.bing.com/");
        URLConnection uc = u.openConnection();
        InputStream stream = uc.getInputStream();
        BufferedInputStream buffer = new BufferedInputStream(stream);
        // chain the InputStream to a Reader
        InputStreamReader reader = new InputStreamReader(buffer);
         int c;
        while ((c = reader.read()) != -1) {
                System.out.print((char) c);
        }

 } catch (IOException ex) {
        System.err.println(ex);
 }
 }
 }
```

2. Write a program to read value of HTTP Header fields

```
import java.io.*;
import java.net.*;
import java.util.*;
public class HeaderViewer {
 public static void main(String[] args) {
try {
 URL u = new URL("https://tufohss.edu.np");
 URLConnection uc = u.openConnection();
 System.out.println("Content-type: " + uc.getContentType());
 if (uc.getContentEncoding() != null) {
        System.out.println("Content-encoding: " + uc.getContentEncoding());
 }
 if (uc.getDate() != 0) {
        System.out.println("Date: " + new Date(uc.getDate()));
 }
 if (uc.getLastModified() != 0) {
        System.out.println("Last modified: " + new Date(uc.getLastModified()));

 }
 if (uc.getExpiration() != 0) {
        System.out.println("Expiration date: " + new Date(uc.getExpiration())); }
 if (uc.getContentLength() != -1) {
        System.out.println("Content-length: " + uc.getContentLength());
```

```
}
} catch (MalformedURLException ex) {
        System.err.println(" is not a URL I understand");
} catch (IOException ex) {
        System.err.println(ex);
}
        System.out.println();
}
}
```

3. Write a program to print the entire HTTP header

```
import java.io.*;
import java.net.*;
public class AllHeaders {
 public static void main(String[] args) {
 for (int i = 0; i < args.length; i++) {
 try {
 URL u = new URL(args[i]);
 URLConnection uc = u.openConnection();
 for (int j = 1; ; j++) {
 String header = uc.getHeaderField(j);
 if (header == null) break;
 System.out.println(uc.getHeaderFieldKey(j) + ": " + header);
 }
 } catch (MalformedURLException ex) {
 System.err.println(args[i] + " is not a URL I understand.");
 } catch (IOException ex) {
 System.err.println(ex);
 }
 System.out.println();
 }
 }
}
```

4. Write a program for HTTP Request Method

HEAD: Get the time when a URL was last changed

```
import java.io.*;
import java.net.*;
import java.util.*;
public class LastModified {
 public static void main(String[] args) {
 try {
        URL u = new URL("https://tufohss.edu.np/");
        HttpURLConnection http = (HttpURLConnection) u.openConnection();
        http.setRequestMethod("HEAD");
        System.out.println(u + " was last modified at "+ new Date(http.getLastModified())); } catch
(MalformedURLException ex) {
        System.err.println(" is not a URL I understand");
} catch (IOException ex) {
        System.err.println(ex);
}
System.out.println();
```

```
}
}
```

5. Write a program to print the URL of a URLConnection to “hafoss.edu.np”

```
import java.io.*;
import java.net.*;
public class URLPrinter {
 public static void main(String[] args) {
 try {
            URL u = new URL("http://www.tufohss.edu.np/");
            URLConnection uc = u.openConnection();
            System.out.println(uc.getURL());
 } catch (IOException ex) {
                System.err.println(ex);
            }
 }
```

6. Write a program to get the time when a URL was last changed

```
import java.io.*;
import java.net.*;
import java.util.*;
public class LastModified {
 public static void main(String[] args) {
 for (int i = 0; i < args.length; i++) {
 try {
 URL u = new URL(args[i]);
 HttpURLConnection http = (HttpURLConnection) u.openConnection();

 http.setRequestMethod("HEAD");
 System.out.println(u + " was last modified at " + new Date(http.getLastModified())); } catch
 (MalformedURLException ex) {
 System.err.println(args[i] + " is not a URL I understand");
 } catch (IOException ex) {
 System.err.println(ex);
 }
 System.out.println();
 }
 }
}
```

Chapter – 6 : Sockets for Clients

1. Write a program socket to client

```
import java.net.*;
import java.io.*;
public class ClientChat {
   public static void main(String[] args) {
      String serverAddress = "127.0.0.1";
      int port = 8888;
      // Try connecting to the server
      try (Socket socket = new Socket(serverAddress, port);
          DataInputStream din = new DataInputStream(socket.getInputStream());
          DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
          BufferedReader br = new BufferedReader(new InputStreamReader(System.in))) {

         System.out.println("Connected to server: " + socket);

         String message;
         do {
            // Read message from console input
            System.out.print("Client: ");
            message = br.readLine();

            // Send message to server
            dout.writeUTF(message);
            dout.flush(); // Ensure the message is sent immediately

            // Read and display the server's response
            String serverMessage = din.readUTF();
            System.out.println("Server: " + serverMessage);

         } while (!message.equalsIgnoreCase("stop"));

      } catch (UnknownHostException e) {
         System.err.println("Server not found: " + e.getMessage());
      } catch (IOException e) {
         System.err.println("I/O error: " + e.getMessage());
      }
   }
}
```

Chapter - 7: Sockets for Servers

1. Write a program socket for a server

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

public class ServerChat {
   public static void main(String[] args) {
      int port = 8888;

      System.out.println("Server START.....");

      // Try-with-resources for automatic closing of server resources
      try (ServerSocket serverSocket = new ServerSocket(port)) {
         System.out.println("Waiting for a client to connect...");
         // Accept the client connection
         try (Socket clientSocket = serverSocket.accept();
              DataInputStream din = new DataInputStream(clientSocket.getInputStream());
              DataOutputStream dout = new DataOutputStream(clientSocket.getOutputStream())) {

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

            String message;
            // Continuously listen to client messages until "stop" is received
            do {
               message = din.readUTF(); // Read client's message
               System.out.println("Client: " + message); // Print client's message

               // Respond back to the client
               dout.writeUTF("Your message is: " + message);
               dout.flush(); // Make sure message is sent immediately
            } while (!message.equalsIgnoreCase("stop")); // Stop when client sends "stop"

            System.out.println("Client disconnected. Server stopping...");

         } catch (IOException e) {
            System.err.println("Error with client connection: " + e.getMessage());
         }
      } catch (IOException e) {
         System.err.println("Server error: " + e.getMessage());
      }
   }
}
```

Chapter – 8 : Secure Sockets

1. Write a program for Creating Secure Sockets with tufohss.edu.np

```
import java.io.*;
import java.net.*;
import javax.net.ssl.*;
public class SecureSocketEx {
 public static void main(String[] a){
 try{
 SSLSocketFactory factory = (SSLSocketFactory) SSLSocketFactory.getDefault();
//System.out.println(factory);
 SSLSocket socket =(SSLSocket) factory.createSocket("tufohss.edu.np", 443);  String[]
supported=socket.getSupportedCipherSuites();
 socket.setEnabledCipherSuites(supported);
 Writer out = new OutputStreamWriter(socket.getOutputStream(),"US-ASCII");
out.write("GET http://tufohss.edu.np/ HTTP/1.1\r\n");
 out.write("Host:tufohss.edu.np \r\n");
 out.write("\r\n");
 out.flush();
        // Read all header fields
 BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));  String s;
 while(!(s=in.readLine()).equals("")){
 System.out.println(s);
 }
 }
 catch(Exception e){}
 }
}
```

2. Write a program for Creating Secure ServerSockets and Client Sockets

ClientSocketEx.java

```
import java.io.*;
import java.net.*;
import javax.net.ssl.*;
public class ClientSocketEx {
 public static void main(String[] a){
 try{
```

```
 Socket socket = ((SSLSocketFactory)SSLSocketFactory.getDefault()).createSocket("localhost",1422);
System.out.println("Server Connected : "+socket); // Connect()
 socket.close();
 }
 catch(Exception e){}
 }
}
```

ServerSocketEx.java

```
import java.io.*;
import java.net.*;
import javax.net.ssl.*;
public class ServerSocketEx {
 public static void main(String[] a){
 try{
 ServerSocket serverSocket =
((SSLServerSocketFactory)SSLServerSocketFactory.getDefault()).createServerSocket(1422);
Socket s=serverSocket.accept();
 System.out.println(s+"Client Accepted and Connected.....");
 }
 catch(Exception e){}
 }
}
```

1. Write program to list all supported socket options for the different types of network channels.

```
import java.io.*;
import java.net.*;
import java.nio.channels.*;
public class OptionSupport {
public static void main(String[] args) throws IOException {
printOptions(SocketChannel.open());
printOptions(ServerSocketChannel.open());
printOptions(AsynchronousSocketChannel.open());
printOptions(AsynchronousServerSocketChannel.open());
printOptions(DatagramChannel.open());
}
private static void printOptions(NetworkChannel channel) throws IOException {
System.out.println(channel.getClass().getSimpleName() + " supports:");
for (SocketOption<?> option : channel.supportedOptions()) {
System.out.println(option.name() + ": " + channel.getOption(option));
}
System.out.println();
channel.close();
}
}
```

2. Write a program to implement the concept on Data Conversion

```
import java.nio.*;
public class DataConversionTest {
public static void main(String[] args) {
// Declaring the capacity of the ByteBuffer (8 bytes = 2 integers)
int capacity = 8;

// Creating a ByteBuffer with the specified capacity
ByteBuffer bb = ByteBuffer.allocate(capacity);

try {
// Putting two integer values into the ByteBuffer using IntBuffer view
bb.asIntBuffer().put(10).put(20);

// Rewind the ByteBuffer to read from the beginning
bb.rewind();

// Print the content of the ByteBuffer (integers)
System.out.println("Original ByteBuffer: ");
for (int i = 1; i <= capacity / 4; i++) { // 4 bytes per int
System.out.print(bb.getInt() + " ");
}

// Rewind the ByteBuffer again to read values step by step
bb.rewind();
```

```
// Reading and printing individual integer values from ByteBuffer
int value1 = bb.getInt();
System.out.println("\n\nFirst Int Value: " + value1);

int value2 = bb.getInt();
System.out.println("Second Int Value: " + value2);

} catch (BufferUnderflowException e) {
System.out.println("\nThere are fewer than four bytes remaining in this buffer.");
System.out.println("Exception Thrown: " + e);
}
}
}
```

1. Write a program for UDP Client example

```
import java.io.*;
import java.net.*;
class UDPClient{
 public static void main(String[] a) throws Exception{
 BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
DatagramSocket clientSocket = new DatagramSocket(); // Socket()
 InetAddress IPAddress = InetAddress.getByName("localhost");
 byte[] sendData = new byte[1024];
 byte[] receiveData = new byte[1024];
 String sentence = inFromUser.readLine();
 sendData = sentence.getBytes();
 DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);
clientSocket.send(sendPacket);

 DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
clientSocket.receive(receivePacket);

 String modifiedSentence = new String(receivePacket.getData());
 System.out.println("FROM SERVER:" + modifiedSentence);
 clientSocket.close();

 }
}
```

2. Write a program for UDP Server example

```
import java.net.*;
import java.io.*;
public class UDPServer {
 public static void main(String[] a) throws Exception{
 DatagramSocket serverSocket = new DatagramSocket(9876); // Socket() and Bind()  byte[]
receiveData = new byte[1024];
 byte[] sendData = new byte[1024];
 while(true)
 {
 DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
serverSocket.receive(receivePacket);
 String sentence = new String( receivePacket.getData());
 System.out.println("RECEIVED: " + sentence);
 InetAddress IPAddress = receivePacket.getAddress();
 int port = receivePacket.getPort();
 sendData = sentence.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress,  port);
 serverSocket.send(sendPacket); // SendTo()
 }

 }
}
```

1. Program to verify that you are receiving multicast data at a particular host

MulticastSocketServer.java

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

public class MulticastSocketServer {
   // Multicast IP address and port
   final static String INET_ADDR = "224.0.0.3";
   final static int PORT = 8888;

   public static void main(String[] args) throws UnknownHostException, InterruptedException {
      // Get the multicast address
      InetAddress addr = InetAddress.getByName(INET_ADDR);

      // Open a DatagramSocket to send the data
      try (DatagramSocket serverSocket = new DatagramSocket()) {
         for (int i = 0; i < 5; i++) {
            String msg = "Sent message no " + i;

            // Create a DatagramPacket with the message and send it
            DatagramPacket msgPacket = new DatagramPacket(msg.getBytes(), msg.length(), addr, PORT);
            serverSocket.send(msgPacket);

            System.out.println("Server sent packet with message: " + msg);

            // Pause for 500 milliseconds between each message
            Thread.sleep(500);
         }
      } catch (IOException ex) {
         ex.printStackTrace();
      }
   }
}
```

MulticastSocketClient.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.UnknownHostException;

public class MulticastSocketClient {
   // Multicast IP address and port
   final static String INET_ADDR = "224.0.0.3";
   final static int PORT = 8888;

   public static void main(String[] args) throws UnknownHostException {
      // Get the multicast address
      InetAddress address = InetAddress.getByName(INET_ADDR);

      // Buffer to store incoming data
      byte[] buf = new byte[256];

      // Open a MulticastSocket to receive the data
      try (MulticastSocket clientSocket = new MulticastSocket(PORT)) {
         // Join the multicast group
         clientSocket.joinGroup(address);

         while (true) {
            // Receive the packet from the multicast group
            DatagramPacket msgPacket = new DatagramPacket(buf, buf.length);
            clientSocket.receive(msgPacket);

            // Convert the received byte array into a string and print it
            String msg = new String(msgPacket.getData(), 0, msgPacket.getLength());
            System.out.println("Client received message: " + msg);
         }
      } catch (IOException ex) {
         ex.printStackTrace();
      }
   }
}
```

Chapter – 12: RMI

1. Program to add two numbers using RMI

Hello.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;
// Creating Remote interface for our application
public interface Hello extends Remote {
        void Add(Integer a,Integer b) throws RemoteException;
}
```

ImplExample.java

```
// Implementing the remote interface
public class ImplExample implements Hello {
   ImplExample(){super();}
  // Implementing the interface method
  public void Add(Integer x, Integer y) {
    System.out.println("Sum = " + (x+y));
  }
}
```

ServerRMI.java

```
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
public class Server extends ImplExample {
 public Server() {}
 public static void main(String args[]) {
 try {
 ImplExample obj = new ImplExample();
Hello skeleton = (Hello) UnicastRemoteObject.exportObject(obj, 0);
Registry registry = LocateRegistry.getRegistry();
 registry.bind("RMITest", skeleton);
System.err.println("Server ready");
 } catch (Exception e) {
 System.err.println("Server exception: " + e.toString());
 e.printStackTrace();

 }
 }
}
```

ClientRMI.java

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class ClientRMI {
 private ClientRMI() {}
 public static void main(String[] args) {
  try {
Registry registry = LocateRegistry.getRegistry();
Hello stub = (Hello) registry.lookup("RMITest");
 stub.Add(5,10);
 } catch (Exception e) {
 System.err.println("Client exception: " + e.toString());
 e.printStackTrace();
 }
```

Conclusion:

In this lab, we learned about the key features of networking, including creating connections between clients and servers, and understanding network security. We also explored how to manage sockets, handle concurrency, ensure security, and optimize performance to build scalable and reliable applications.