Network Programming Lab Report

Year/Semester: III / VI

Index

18	HTTP Request Method		
19	URLConnection to “nepathyacollege.edu.np”		
20	Get last modified time of a URL		
Chapter 6: Sockets for Clients			
21	Socket Program for Client		
Chapter 7: Sockets for Server			
22	Socket Program for Server		
Chapter 8: Secure Sockets			
23	Create Secure Sockets with tufohss.edu.np		
24	Create Secure ServerSockets and ClientSockets		
Chapter 9: Non-Blocking I/O			
25	List supported socket options for various network channels		
26	Buffer operations (Fill, Drain, Duplicate, Slice, Compact)		
27	Data Conversion		
Chapter 10: UDP			
28	UDP Client		
29	UDP Server		
Chapter 11: IP Multicast			
30	Verify reception of multicast data at a host		
Chapter 12: RMI			
31	Add two numbers using RMI		

LAB-1

AIM: Write a program to check IPv4 and IPv6 Address.

Program:

```
import java.net.*;
public class IpTypeCheck{
   public static void main(String[] args) {
      String host = "bhuvanbhusal.com.np";
      try {
         InetAddress address = InetAddress.getByName(host);
         byte[] ipBytes = address.getAddress();
         if (ipBytes.length == 4) {
            System.out.println(host + " is IPv4.");
         } else if (ipBytes.length == 16) {
            System.out.println(host + " is IPv6.");
         } else {
            System.out.println("Unknown format.");
         }
      } catch (UnknownHostException e) {
         System.out.println("Invalid host: " + host);
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVir
bhuvanbhusal.com.np is IPv4.

Process finished with exit code 0
```

LAB-2

AIM: Write a program that finds the address of the local machine

Program:

```
import java.net.*;
public class LocalMachineAddress {
   public static void main(String[] args) {
      try {
         InetAddress localhost = InetAddress.getLocalHost();
         System.out.println("Local machine address: " + localhost.getHostAddress());
      } catch (UnknownHostException e) {
         System.err.println("Could not determine local host address: " + e.getMessage());
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVir
Local machine address: 127.0.0.1

Process finished with exit code 0
```

LAB-3

AIM: Write a program to find the IP address and Host name of the local machine

Program:

```
import java.net.*;


public class LocalHostInfo {
   public static void main(String[] args) {
      try {
         InetAddress localhost = InetAddress.getLocalHost();
         System.out.println("Host Name: " + localhost.getHostName());
         System.out.println("IP Address: " + localhost.getHostAddress());
      } catch (UnknownHostException e) {
         System.err.println("Could not determine local host information: " + e.getMessage());
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVir
Host Name: Alexs-MacBook-Air.local
IP Address: 127.0.0.1

Process finished with exit code 0
```

LAB-4

AIM: Write a program that demonstrate the SpamCheck.

Program:

```
import java.net.*;
public class SpamCheck {
   public static void main(String[] args) {
      try {
         InetAddress address = InetAddress.getByName("www.bhuvanbhusal.com.np");
         if (!address.isAnyLocalAddress()) {
            System.out.println("May be Spam Address");
         } else {
            System.out.println("Safe Address");
         }
      } catch (Exception e) {
         System.out.println("Error :" + e.getMessage());
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java
May be Spam Address
```

LAB-5

AIM: Write a program that compare the domain name are "www.ibiblio.org" and "helios.ibiblio.org" the same?

Program:

```
import java.net.*;

public class DomainCompare {
    public static void main(String[] args) {
        try {
            InetAddress address1 = InetAddress.getByName("www.ibiblio.org");
            InetAddress address2 = InetAddress.getByName("helios.ibiblio.org");


            if (address1.equals(address2)) {
                System.out.println("Both domains point to the same address.");
            } else {
                System.out.println("The domains are different.");
            }
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/
The domains are different.

Process finished with exit code 0
```

LAB-6

AIM: Write a program that splits the parts of a URL [Splitting URL into pieces information]

Program:

```
import java.net.*;

public class URLSplitter {

    public static void main(String[] args) {

        try {

            URL url = new URL("https://www.example.com:8080/path/to/resource?
            query=123&lang=en");

            System.out.println("Protocol: " + url.getProtocol());

            System.out.println("Host: " + url.getHost());

            System.out.println("Port: " + url.getPort());

            System.out.println("Path: " + url.getPath());

            System.out.println("Query: " + url.getQuery());

            System.out.println("File: " + url.getFile());

        } catch (Exception e) {

            System.out.println("Error: " + e.getMessage());

        }

    }

}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachi
Protocol: https
Host: www.example.com
Port: 8080
Path: /path/to/resource
Query:  query=123&lang=en
File: /path/to/resource?    query=123&lang=en
```

LAB-7

AIM: Write a program that checks the which protocols does a virtual machine support or not?

Program:

```
import java.net.*;

public class ProtocolChecker {
    public static void main(String[] args) {
        String[] protocols = {"http", "https", "ftp", "mailto", "file", "jdbc", "telnet"};
        for (String protocol : protocols) {
            try {
                URL url = new URL(protocol, "example.com", "");
                System.out.println(protocol + " is supported");
            } catch (Exception e) {
                System.out.println(protocol + " is NOT supported: " + e.getMessage());
            }
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/
http is supported
https is supported
ftp is supported
mailto is supported
file is supported
jdbc is NOT supported: unknown protocol: jdbc
telnet is NOT supported: unknown protocol: telnet
```

LAB-8

AIM: Write a program to download a web page of a given web address.

Program:

```
import java.io.*;
public class WebPageDownloader {
    public static void main(String[] args) {
        String urlString = "https://www.example.com";
        String outputFile = "downloaded_page.html";
        try (InputStream inputStream = new URL(urlString).openStream();
             FileOutputStream fileOutputStream = new FileOutputStream(outputFile)) {
            byte[] buffer = new byte[1024];
            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                fileOutputStream.write(buffer, 0, bytesRead);
            }
            System.out.println("Web page downloaded successfully to " + outputFile);
        } catch (IOException e) {
            System.err.println("Error downloading the web page: " + e.getMessage());
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-
Web page downloaded successfully to downloaded_page.html
```

LAB-9

AIM: Write a program for resolving relatives URI

Program:

```
import java.net.*
public class ResolveURIExample {
    public static void main(String[] args) {
        try {
            URI baseUri = new URI("http://example.com/path/to/resource/");
            URI relativeUri = new URI("subdir/resource2");
            URI resolvedUri = baseUri.resolve(relativeUri);
            System.out.println("Resolved URI: " + resolvedUri);
        } catch (URISyntaxException e) {
            System.err.println("Invalid URI syntax: " + e.getMessage());
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Contents
Resolved URI: http://example.com/path/to/resource/subdir/resource2

Process finished with exit code 0
```

LAB-10

AIM: Write a program to download an object.

Program:

```
import java.*;
public class DownloadObjectExample {
   public static void main(String[] args) {
      try {
         // Open the URL for reading
         URL u = new URL("http://www.google.com");
         Object o = u.getContent();
         System.out.println("I got a " + o.getClass().getName());
      } catch (MalformedURLException ex) {
         System.err.println("URL is not parseable");
      } catch (IOException ex) {
         System.err.println("Error retrieving content from the URL");
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Contents/
I got a sun.net.www.protocol.http.HttpURLConnection$HttpInputStream
```

LAB-11

AIM: Write a program to demonstrate the x-www-form-URL encoded strings.

Program:

```
import java.net.*;

public class FormURLEncoder {
    public static void main(String[] args) {
        String original = "Hello World! @Java Programming?name=value&test=123";
        try {
            String encoded = URLEncoder.encode(original, "UTF-8");
            String decoded = URLDecoder.decode(encoded, "UTF-8");
            System.out.println("Original: " + original);
            System.out.println("Encoded: " + encoded);
            System.out.println("Decoded: " + decoded);
        } catch (UnsupportedEncodingException e) {
            System.err.println("UTF-8 encoding not supported");
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Contents/Home
Original: Hello World! @Java Programming?name=value&test=123
Encoded: Hello+World%21+%40Java+Programming%3Fname%3Dvalue%26test%3D123
Decoded: Hello World! @Java Programming?name=value&test=123
```

LAB-12

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

Program:

```
import java.io.*;
import java.net.*;
public class GetRequestExample {
    public static void main(String[] args) throws Exception {
        BufferedReader in = new BufferedReader(new InputStreamReader(
            new URL("http://httpbin.org/get?param1=value1&param2=value2").openStream()));
        in.lines().forEach(System.out::println);
        in.close();
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Contents/Home/
Sending GET request to: http://httpbin.org/get?param1=value1&param2=value2
Response:
{
  "args": {
    "param1": "value1",
    "param2": "value2"
  },
  "headers": {
    "Accept": "*/*",
    "Host": "httpbin.org",
    "User-Agent": "Java/24",
    "X-Amzn-Trace-Id": "Root=1-68149c73-75fc60922d9418c45a5fd5a8"
  },
  "origin": "103.123.60.240",
  "url": "http://httpbin.org/get?param1=value1&param2=value2"
}

Process finished with exit code 0
```

LAB-13

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

Program:

```
import java.net.*;
public class NoGovernmentCookies implements CookiePolicy {
   @Override
   public boolean shouldAccept(URI uri, HttpCookie cookie) {
      return !uri.getAuthority().endsWith(".gov") && !cookie.getDomain().endsWith(".gov");
   }
   public static void main(String[] args) {
      NoGovernmentCookies policy = new NoGovernmentCookies();
      try {
         System.out.println(policy.shouldAccept(new URI("http://example.gov"),
          new HttpCookie("sessionid", "12345").setDomain("example.gov")));
         System.out.println(policy.shouldAccept(new URI("http://example.com"),
          new HttpCookie("sessionid", "67890").setDomain("example.com")));
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/
false
true
```

LAB-14

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

Program:

```
import java.*;
public class CookieStoreExample {
    public static void main(String[] args) throws Exception {
        // Create a CookieManager with a CookieStore
        CookieManager manager = new CookieManager();
        CookieHandler.setDefault(manager);
        CookieStore cookieStore = manager.getCookieStore();

        // Add cookies
        URI uri1 = new URI("http://example.com");
        HttpCookie cookie1 = new HttpCookie("username", "john_doe");
        cookieStore.add(uri1, cookie1);
        URI uri2 = new URI("http://test.org");
        HttpCookie cookie2 = new HttpCookie("session", "abc123");
        cookieStore.add(uri2, cookie2);

        // Read cookies
        System.out.println("All cookies in store:");
        List<HttpCookie> cookies = cookieStore.getCookies();
        for (HttpCookie cookie : cookies) {
            System.out.println(cookie);
       }
        // Get cookies for specific URI
        System.out.println("\nCookies for example.com:");
        List<HttpCookie> exampleCookies = cookieStore.get(uri1);
```

```
        for (HttpCookie cookie : exampleCookies) {
            System.out.println(cookie);
        }
        // Delete a cookie
        System.out.println("\nDeleting session cookie...");
        cookieStore.remove(uri2, cookie2);
        System.out.println("\nRemaining cookies:");
        cookies = cookieStore.getCookies();
        for (HttpCookie cookie : cookies) {
            System.out.println(cookie);
        }
        // Delete all cookies
        System.out.println("\nClearing all cookies...");
        cookieStore.removeAll();
        System.out.println("Cookies after clearing: " + cookieStore.getCookies().size());
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtual
All cookies in store:
username="alex_bhusal"
session="alex9876"

Cookies for example.com:
username="alex_bhusal"

Deleting session cookie...

Remaining cookies:
username="alex_bhusal"

Clearing all cookies...
Cookies after clearing: 0

Process finished with exit code 0
```

LAB-15

AIM: Write a program to download a web page using URLConnection.

Program:

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


public class WebPageDownloader {
   public static void main(String[] args) throws IOException {
      BufferedReader in = new BufferedReader(new InputStreamReader(
         new URL("https://www.example.com").openStream()));
      in.lines().forEach(System.out::println);
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Conte
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf
    <meta name="viewport" content="width=device-width, initial-scal
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont,

    }
    div {
        width: 600px;
```

LAB-16

AIM: Write a program to read value of HTTP Header fields.

Program:

```
import java.io*;
public class HeaderViewer {
   public static void main(String[] args) {
      try {
         URLConnection uc = new URL("https://tufohss.edu.np").openConnection();
         System.out.println("Content-type: " + uc.getContentType());
         System.out.println("Content-encoding: " + uc.getContentEncoding());
         System.out.println("Date: " + new Date(uc.getDate()));
         System.out.println("Last modified: " + new Date(uc.getLastModified()));
         System.out.println("Expiration date: " + new Date(uc.getExpiration()));
         System.out.println("Content-length: " + uc.getContentLength());
      } catch (IOException ex) {
         System.err.println(ex);
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/o
Content-type: text/html; charset=iso-8859-1
Content-encoding: null
Date: Fri May 02 19:06:09 GMT+05:45 2025
Last modified: Thu Jan 01 05:30:00 GMT+05:45 1970
Expiration date: Thu Jan 01 05:30:00 GMT+05:45 1970
Content-length: 231
```

LAB-17

AIM: Write a program to print the entire HTTP header

Program:

```
import java.io.*;
import java.net.*;
public class HttpHeaderPrinter {
    public static void main(String[] args) throws IOException {
        HttpURLConnection connection = (HttpURLConnection)
            new URL("https://www.example.com").openConnection();
        connection.setRequestMethod("GET");
        connection.connect();
        connection.getHeaderFields().forEach((k, v) ->
            System.out.println(k + ": " + String.join(", ", v)));
        connection.disconnect();
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Contents/Home/bin/java -
null: HTTP/1.1 200 OK
Cache-Control: max-age=805
Alt-Svc: h3=":443"; ma=93600,h3-29=":443"; ma=93600,quic=":443"; ma=93600; v="43"
ETag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
Connection: keep-alive
Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT
Content-Length: 1256
Date: Fri, 02 May 2025 10:37:10 GMT
Content-Type: text/html
```

LAB-18

AIM: Write a program for HTTP Request Method

Program:

```
import java.net.*;

public class HttpRequestExample {

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

      URL url = new URL(" https://example.com");

      HttpURLConnection connection = (HttpURLConnection) url.openConnection();

      connection.setRequestMethod("GET");

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

      String inputLine;

      StringBuffer response = new StringBuffer();

      while ((inputLine = in.readLine()) != null) {

         response.append(inputLine);

      }

      in.close();

      System.out.println(response.toString());

   }

}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Conte
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf
    <meta name="viewport" content="width=device-width, initial-scal
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont,

    }
    div {
        width: 600px;
```

LAB-19

AIM: Write a program to print the URL of a URL Connection to "nepathyacollege.edu.np".

Program:

```
import java.net.*;

public class URLPrinter {
    public static void main(String[] args) {
        String urlString = "http://nepathyacollege.edu.np";
        try {
            URL url = new URL(urlString);
            URLConnection connection = url.openConnection();
            System.out.println("URL of the connection: " + connection.getURL());
        } catch (IOException e) {
            System.err.println("Error opening connection: " + e.getMessage());
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/
URL of the connection: http://nepathyacollege.edu.np
```

LAB-20

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

Program:

```
import java.*;
public class URLLastModified {
   public static void main(String[] args) {
      String urlString = "http://example.com";

      try {
         URL url = new URL(urlString);
         URLConnection connection = url.openConnection();
         long lastModified = connection.getLastModified();
         if (lastModified == 0) {
            System.out.println("Last modified information not available");
         } else {
            System.out.println("Last modified: " + new Date(lastModified));
         }
      } catch (IOException e) {
         System.err.println("Error getting last modified time: " + e.getMessage());
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/
Last modified: Tue Jan 14 01:56:20 GMT+05:45 2025
```

LAB-21

AIM: Write a program socket to client.

Program:

```
import java.*;
public class Client {
   public static void main(String[] args) throws IOException {
      Socket socket = new Socket("localhost", 1234);
      PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
      BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
      out.println("Hello Server");
      String response = in.readLine();
      System.out.println("Server says: " + response);
      socket.close();
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/J
Server says: Hello Client
```

LAB-22

AIM: Write a program socket for a server.

Program:

```
import java.net.*;
public class Server {
   public static void main(String[] args) throws IOException {
      ServerSocket serverSocket = new ServerSocket(1234);
      System.out.println("Server is listening on port 1234...");
      Socket socket = serverSocket.accept();
      BufferedReader in = new BufferedReader(new
         InputStreamReader(socket.getInputStream()));
      PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
      String clientMessage = in.readLine();
      System.out.println("Client says: " + clientMessage);
      out.println("Hello Client");
      socket.close();
      serverSocket.close();
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualM
Server is listening on port 1234...
Client says: Hello Server
```

LAB-23

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

Program:

```
import java.net.*;
public class SecureSocketClient {
   public static void main(String[] args) {
      try (SSLSocket socket = (SSLSocket) SSLSocketFactory.getDefault().
         createSocket("tufohss.edu.np", 443)) {
         socket.setEnabledCipherSuites(socket.getSupportedCipherSuites());
         PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
         out.println("GET / HTTP/1.1\nHost: tufohss.edu.np\n");
         new BufferedReader(new InputStreamReader(socket.getInputStream()))
            .lines().forEach(System.out::println);
      } catch (IOException e) {
         System.err.println("Error: " + e.getMessage());
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines/openjdk-24/Contents/Home/bin/java --
HTTP/1.1 400 Bad Request
Date: Fri, 02 May 2025 13:32:01 GMT
Server: Apache
Content-Length: 347
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<p>Additionally, a 400 Bad Request
error was encountered while trying to use an ErrorDocument to handle the request.</p>
</body></html>
```

LAB-24

AIM: Write a program for Creating Secure ServerSockets and Client Sockets

Program:

```
import java.net.*;
public class SecureServer {
   public static void main(String[] args) throws Exception {
      System.setProperty("javax.net.ssl.keyStore", "serverKeyStore.jks");
      System.setProperty("javax.net.ssl.keyStorePassword", "password");
      SSLServerSocket serverSocket = (SSLServerSocket)
         SSLServerSocketFactory.getDefault().createServerSocket(12345);
      while (true) {
         SSLSocket socket = (SSLSocket) serverSocket.accept();
         new PrintWriter(socket.getOutputStream(), true).println("Hello from server");
         socket.close();
      }
   }
}
import javax.*;
public class SecureClient {
   public static void main(String[] args) throws Exception {
      System.setProperty("javax.net.ssl.trustStore", "clientTrustStore.jks");
      System.setProperty("javax.net.ssl.trustStorePassword", "password");
      SSLSocket socket = (SSLSocket)
         SSLSocketFactory.getDefault().createSocket("localhost", 12345);
      System.out.println(new BufferedReader(new
InputStreamReader(socket.getInputStream())).readLine());
      socket.close();
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualI
Secure server started on port 12345
Client connected: /127.0.0.1
Received: Hello, Secure Server!
```

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines
Server response: Echo: Hello, Secure Server!
```

LAB-25

AIM: Write program to list all supported socket options for the different types of network channels

Program:

```
import java.*;
public class SocketOptionsLister {
   public static void main(String[] args) throws IOException {
      for (Channel channel : new Channel[] {
         ServerSocketChannel.open(), SocketChannel.open(), DatagramChannel.open() }) {
         System.out.println(channel.getClass().getSimpleName() + " supported options:");
         channel.supportedOptions().forEach(option -> System.out.println(option.name()));
         System.out.println();
      }
   }
}
```

Output:

```
ServerSocketChannelImpl supported options:
TCP_KEEPINTERVAL
SO_RCVBUF
SO_REUSEPORT
TCP_KEEPCOUNT
SO_REUSEADDR
TCP_KEEPIDLE

SocketChannelImpl supported options:
TCP_KEEPINTERVAL
SO_SNDBUF
SO_RCVBUF
SO_REUSEPORT
TCP_KEEPCOUNT
SO_REUSEADDR
SO_LINGER
TCP_NODELAY
SO_KEEPALIVE
IP_TOS
TCP_KEEPIDLE
SO_OOBINLINE

DatagramChannelImpl supported options:
SO_SNDBUF
SO_RCVBUF
SO_REUSEPORT
SO_REUSEADDR
IP_MULTICAST_IF
IP_DONTFRAGMENT
IP_MULTICAST_LOOP
```

LAB-26

AIM: Write program to implement the concept on Filling and Draining buffer, duplicating buffer, Slicing buffer, Compact buffer.

Program:

```
import java.nio.*;

public class BufferExamples {

    public static void main(String[] args) {

        ByteBuffer buffer = ByteBuffer.allocate(10);

        // Filling & Draining

        buffer.put((byte) 1).put((byte) 2).flip();

        System.out.println(buffer.get() + ", " + buffer.get());

        // Duplicating

        System.out.println(buffer.duplicate());

        // Slicing

        buffer.clear().put((byte) 3).put((byte) 4).flip();

        System.out.println(buffer.slice());

        // Compacting

        buffer.clear().put((byte) 5).put((byte) 6).flip();

        buffer.get().compact();

        System.out.println(buffer);

    }

}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines
Filling & Draining:
1, 2

Duplicating:
Duplicate buffer position: 2, limit: 2

Slicing:
Slice position: 0, limit: 2
Slice content: 3, 4

Compacting:
Buffer after compact: position = 1, limit = 10
6
```

LAB-27

AIM: Write a program to implement the concept on Data Conversion

Program:

```
import java.*;
public class DataConversion {
   public static void main(String[] a) {
      ByteBuffer b = ByteBuffer.allocate(16);
      b.putInt(42).putDouble(3.14).putChar('A').flip();
      System.out.println(b.getInt() + " " + b.getDouble() + " " + b.getChar());
      ByteBuffer sb = StandardCharsets.UTF_8.encode("Hello NIO");
      System.out.println( StandardCharsets.UTF_8.decode((ByteBuffer) sb.rewind()));
      ByteBuffer src = ByteBuffer.allocate(16);
      for (int i = 0; i < 16; i++) src.put((byte) i);
      System.out.print("\n");
      src.flip();
      IntBuffer ib = src.asIntBuffer();
      while (ib.hasRemaining()) System.out.print(ib.get() + " ");
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVirtualMachines
Filling & Draining:
1, 2

Duplicating:
Duplicate buffer position: 2, limit: 2

Slicing:
Slice position: 0, limit: 2
Slice content: 3, 4

Compacting:
Buffer after compact: position = 1, limit = 10
6
```

LAB-28

AIM: Write a program for UDP Client.

Program:

```
import java.net.*;
public class UDPClient {
   public static void main(String[] args) {
      try (DatagramSocket socket = new DatagramSocket()) {
         byte[] buf = "Hello UDP Server".getBytes();
         InetAddress address = InetAddress.getByName("localhost");
         socket.send(new DatagramPacket(buf, buf.length, address, 9876));
         byte[] respBuf = new byte[1024];
         DatagramPacket response = new DatagramPacket(respBuf, respBuf.length);
         socket.receive(response);
         System.out.println("Server response: " + new String(response.getData(), 0,
response.getLength()));
      } catch (Exception e) {
         System.err.println("Error: " + e.getMessage());
      }
   }
}
```

Output:

```
/Users/alexbhusal/Library/Java/JavaVir
Server response: Hello from Server
```

LAB-29

AIM: Write a program for UDP Server.

Program:

```
 import java.net.*;
public class UDPServer {
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket(9876)) {
            byte[] buffer = new byte[1024];
            System.out.println("Server running...");
            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            socket.receive(request);
            String msg = new String(request.getData(), 0, request.getLength());
            System.out.println("Received: " + msg);
            byte[] reply = "Hello from Server".getBytes();
            socket.send(new DatagramPacket(reply, reply.length, request.getAddress(),
request.getPort()));
        } catch (Exception e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/
Server running...
Received: Hello UDP Server
```

LAB-30

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

Program:

```
import java.net.*;

public class MulticastReceiver {

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

        String groupAddress = "224.0.0.1"; // Multicast address

        int port = 5000; // Multicast port

        MulticastSocket socket = new MulticastSocket(port);

        InetAddress group = InetAddress.getByName(groupAddress);

        socket.joinGroup(group);

        byte[] buffer = new byte[1024];

        while (true) {

            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);

            socket.receive(packet);

            System.out.println("Received: " + new String(packet.getData(), 0,
packet.getLength()));

        }

    }

}
```

Output:

```
/Users/alexbhusal/Library/Java
Received: Hello Multicast
```

LAB-31

AIM: Program to add two numbers using RMI

Program:

// Remote Interface (Adder.java)

```
import java.*;
public interface Adder extends Remote {
   int add(int num1, int num2) throws RemoteException;
}
```

//Remote Implementation (AdderImpl.java)

```
import java.*;
public class AdderImpl extends UnicastRemoteObject implements Adder {
   public AdderImpl() throws RemoteException {}
   public int add(int num1, int num2) throws RemoteException {
      return num1 + num2;
   }
}
```

//Server (AdderServer.java)

```
import java.*;
public class AdderServer {
   public static void main(String[] args) throws Exception {
      LocateRegistry.createRegistry(1099);
      Naming.rebind("AdderService", new AdderImpl());
      System.out.println("Server ready.");
   }
}
```

//Client (AdderClient.java)

```
import java.*;
public class AdderClient {
    public static void main(String[] args) throws Exception {
        Adder adder = (Adder) Naming.lookup("rmi://localhost/AdderService");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
        int num1 = sc.nextInt(), num2 = sc.nextInt();
        System.out.println("Sum: " + adder.add(num1, num2));
    }
}
```

Output:

```
/Users/alexbhusal/Library/Java/
Server ready.
```

```
/Users/alexbhusal/Library/Java/
Enter two numbers: 100
200
Sum: 300
```