Java Lab 8

Network Programming Basic
Compiled by : Bidur Devkota

1# Program to get the IP address of given host.

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

public class IP
{
  public static void main ( String[] args ) throws IOException
  {
    String hostname = args[0];

    try
    {
      InetAddress ipaddress = InetAddress.getByName(hostname);
      System.out.println("IP address: " + ipaddress.getHostAddress());
    }
    catch ( UnknownHostException e )
    {
      System.out.println("Could not find IP address for: " + hostname);
    }
  }
}

// compile: javac IP.java
//run: java IP google.com
```

2# The following URLDemo program demonstrates the various parts of a URL.
A URL is entered on the command line, and the URLDemo program outputs
each part of the given URL.

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

public class UrlDemoJava
{
   public static void main(String [] args)
   {
      try
      {
```

```
      //  URL url = new URL("https://www.google.com.np/?gws_rd=cr&ei=URmRU8y_EcfpkAWK4oDoCA");
          URL url = new URL("https://www.gces.edu.np");
         System.out.println("URL is " + url.toString());
         System.out.println("protocol is "
                                    + url.getProtocol());
         System.out.println("authority is "
                                    + url.getAuthority());
         System.out.println("file name is " + url.getFile());
         System.out.println("host is " + url.getHost());
         System.out.println("path is " + url.getPath());
         System.out.println("port is " + url.getPort());
         System.out.println("default port is "
                                   + url.getDefaultPort());
         System.out.println("query is " + url.getQuery());
         System.out.println("ref is " + url.getRef());
      }catch(IOException e)
      {
         e.printStackTrace();
      }
   }
}


3# The Program to download the content of the given url.

import java.net.*;
import java.io.*;
public class UrlConnectionDemoJava
{
   public static void main(String [] args)
   {
      try
      {
         URL url = new URL("http://www.google.com");
         URLConnection urlConnection = url.openConnection();
         HttpURLConnection connection = null;
/*instanceof keyword can be used to test if an object is of a specified type.
if (objectReference instanceof type
*/
         if(urlConnection instanceof HttpURLConnection)
         {
            connection = (HttpURLConnection) urlConnection;
         }
         else
         {
            System.out.println("Please enter an HTTP URL.");
            return;
         }
         BufferedReader in = new BufferedReader(
         new InputStreamReader(connection.getInputStream()));
         String urlString = "";
         String current;
         while((current = in.readLine()) != null)
         {
            urlString += current;
         }
         System.out.println(urlString);
      }catch(IOException e)
      {
         e.printStackTrace();
      }
   }
}

4# Program to demonstrate client server communication.

// SimpleServer.java: A simple server program.
import java.net.*;
import java.io.*;
public class SimpleServer {
    public static void main(String args[]) throws IOException {
        // Register service on port 1254
        ServerSocket s = new ServerSocket(1254);
        Socket s1=s.accept(); // Wait and accept a connection
        // Get a communication stream associated with the socket
        OutputStream s1out = s1.getOutputStream();
        DataOutputStream dos = new DataOutputStream (s1out);
        // Send a string!
        dos.writeUTF("Hi there");
        // Close the connection, but not the server socket
        dos.close();
        s1out.close();
        s1.close();
    }
}
```

```
// SimpleClient.java: A simple client program.
import java.net.*;
import java.io.*;
public class SimpleClient {
    public static void main(String args[]) throws IOException {
        // Open your connection to a server, at port 1254
        Socket s1 = new Socket("localhost",1254);
        // Get an input file handle from the socket and read the input
        InputStream s1In = s1.getInputStream();
        DataInputStream dis = new DataInputStream(s1In);
        String st = new String (dis.readUTF());
        System.out.println(st);
        // When done, just close the connection and exit
        dis.close();
        s1In.close();
        s1.close();
    }
}


5# Use UDP tp write a program to demonstrate cleint server communication
where a client program reads some message from the user and sends it to
the server. Then the server sends acknoledge to the client.

6# Use TCP tp write a program to demonstrate cleint server communication
 where a client program reads some message from the user and sends it to
 the server. Then the server sends acknoledge to the client.
```