Q1.

There are many protocols in the application layer. The most common protocols are FTP, SMTP, TELNET, SSH, HTTP, and DNS.

FTP: this is a standard internet protocol for the transfer of files from one computer to another, which typically uses port 21. A username and password are usually required to access ftp servers.

SMTP: This is a standard TCP/IP protocol used in sending and receiving e-mail, which typically uses pot 525

TELNET: This is a standard protocol for remote login on devices, however it is very unsecure. It utilizes port 23.

SSH: This is a standard protocol for remote login on devices, however it Is much more secure than telnet. It utilizes 22.

HTTP: This is a standard TCP/IP protocol used in delivering data on the world wide web. It utilizes port 80.

DNS: This is a standard protocol for the translation of domain names into IP addresses. DNS is like the phone book of the internet.

Q2.

Client

import java.net.\*;  
import java.io.\*;  
import java.util.Scanner;  
public class Client {  
 public static void main(String[] args) {  
try {  
Socket client = new Socket( "localhost",5000 );  
System.*out*.println("Client is connected to Server");  
ObjectOutputStream out = new ObjectOutputStream( client.getOutputStream() );  
ObjectInputStream in = new ObjectInputStream( client.getInputStream() );  
Scanner input = new Scanner( System.*in*);  
while (true) {  
System.*out*.println("Enter your Message");  
String message = input.nextLine();  
out.writeObject(message);  
message = ( String ) in.readObject();  
System.*out*.println ("Message from Server: " + message);  
if (message.equals("bye")) {  
 System.*out*.println("Shutting down");  
 break;  
}  
}  
}  
catch (Exception ex) {  
System.*err*.println(ex);  
}  
}  
}

Server

import java.net.\*;  
import java.io.\*;  
import java.lang.Math;  
public class Server {  
public static void main(String[] args) {  
try {  
ServerSocket server = new ServerSocket( 5000, 10 );  
System.*out*.println("Server is Running on port 4000");  
Socket connection = server.accept();  
ObjectOutputStream out = new ObjectOutputStream( connection.getOutputStream() );  
ObjectInputStream in = new ObjectInputStream( connection.getInputStream() );  
String message = "Connection successful";  
boolean end = false;  
String username = "DIT";  
String password = "TUDUBLIN";  
while (!end) {  
message = ( String ) in.readObject();  
try {  
switch(message.toLowerCase()) {  
case "register":  
out.writeObject("Please send your name");  
in.readObject();  
out.writeObject("Please send your age");  
in.readObject();  
out.writeObject("Please send your phone number");  
in.readObject();  
out.writeObject("The Client is successfully registered");  
break;  
case "login":  
out.writeObject("Please enter your username");  
if (!(((String) in.readObject()).equals(username))) {  
out.writeObject("Invalid username");  
break;  
}  
out.writeObject("Please enter your password");  
if (((String) in.readObject()).equals(password)) {  
out.writeObject("You are successfully logged in");  
}  
else {  
out.writeObject("Incorrect password");  
}  
break;  
case "modify":  
out.writeObject("Please send a new password");  
password = (String) in.readObject();  
out.writeObject("New password is "+password);  
break;  
case "bye":  
end = true;  
break;  
default:  
out.writeObject("Invalid input, please try again");  
}  
}  
catch (Exception ex) {  
 out.writeObject("Invalid input");  
}  
}  
out.writeObject("bye");  
System.*out*.println("Shutting down");  
}  
catch (Exception ex) {  
System.*err*.println(ex);  
}  
}  
}

Q3

Client

package question.three;  
  
import java.io.\*;  
import java.net.\*;  
import java.net.InetAddress;  
import org.apache.commons.net.daytime.DaytimeUDPClient;  
class MyUDPClient{  
public static void main(String[]args){  
try{  
DaytimeUDPClient client = new DaytimeUDPClient();  
InetAddress address = InetAddress.*getByName*("time.nist.gov");  
client.setDefaultTimeout(60000);  
client.open();  
System.*out*.println("Getting the Time from UDP Server");  
System.*out*.println(client.getTime(address,13));}  
catch(Exception e){  
System.*out*.println(e);  
}  
}  
}

Server

package question.three;  
  
import java.io.\*;  
import java.net.\*;  
import java.net.InetAddress;  
import org.apache.commons.net.daytime.DaytimeUDPClient;  
public class DaytimeServer {  
 public static void main(String args[ ]) throws java.io.IOException {  
 String host = args[0];  
 int port = 13;  
 if (args.length > 1) port = Integer.*parseInt*(args[1]);  
 DatagramSocket socket = new DatagramSocket( );  
 socket.setSoTimeout(1000);  
 byte[ ] buffer = new byte[512];  
 DatagramPacket packet = new DatagramPacket(buffer, buffer.length,  
 new InetSocketAddress(host,port));  
 for(int i = 0; i < 3; i++) {  
 try {  
 // Send an empty datagram to the specified host (and port)  
 packet.setLength(0);  
 socket.send(packet);  
 packet.setLength(buffer.length);  
 socket.receive(packet);  
 System.*out*.print(new String(buffer, 0, packet.getLength( ),  
 "US-ASCII"));  
 break;  
 }  
 catch(SocketTimeoutException e) {  
 System.*out*.println("No response");  
 }  
 }  
 socket.close( );  
 }  
 }

Q4

Client

public class TCPClient {  
public static void main(String[] args) {  
Scanner scanner = new Scanner(System.*in*);  
  
System.*out*.println("1. Scan all ports 2. Get Time 3. Get domain Info");  
System.*out*.println("Enter a number: ");  
  
int input = scanner.nextInt();  
  
switch (input) {  
case 1:  
for (int port = 1; port <= 65535; port++) {  
try {  
 Socket socket = new Socket("127.0.0.1", port);  
 System.*out*.println("Port in use: " + port);  
 socket.close();  
} catch (Exception exception) {  
 System.*out*.println("Port not in use: " + port);  
}  
}  
  
case 2:  
try {  
Socket socket = new Socket("time.nist.gov", 13);  
socket.setSoTimeout(15000);  
InputStream in = socket.getInputStream();  
int c;  
while ((c = in.read()) != -1) {  
 System.*out*.print((char) c);  
}  
socket.close();  
} catch (Exception exception) {  
System.*err*.println(exception);  
}  
case 3:  
try {  
String domain = scanner.nextLine();  
Socket socket = new Socket("whois.internic.net", 43);  
InputStream in = socket.getInputStream();  
OutputStream out = socket.getOutputStream();  
out.write(domain.getBytes());  
int c;  
while ((c = in.read()) != -1) {  
 System.*out*.print((char) c);  
}  
socket.close();  
} catch (Exception exception) {  
System.*err*.println(exception);  
}  
}  
}  
}

s