Networking and Programming Assignments (Java & C)

1. Java Client-Server Program for Current Date & Time

```
Server Code (DateTimeServer.java):
---------------------------------
import java.io.*;
import java.net.*;
import java.util.Date;

public class DateTimeServer {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(5000);
            System.out.println("Server is running...");
            while (true) {
                Socket clientSocket = serverSocket.accept();
                PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
                Date currentDate = new Date();
                out.println("Current Date and Time: " + currentDate.toString());
                clientSocket.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

Client Code (DateTimeClient.java):
---------------------------------
import java.io.*;
import java.net.*;

public class DateTimeClient {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 5000);
                                    BufferedReader   in   =   new   BufferedReader(new
InputStreamReader(socket.getInputStream()));
            String serverResponse = in.readLine();
            System.out.println("Server says: " + serverResponse);
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

2. Java Program to Access IP Address and Class

```
Code (IPClassIdentifier.java):
```

```
------------------------------
import java.net.*;

public class IPClassIdentifier {
    public static void main(String[] args) {
        try {
            InetAddress localHost = InetAddress.getLocalHost();
            String ipAddress = localHost.getHostAddress();
            System.out.println("Local IP Address: " + ipAddress);
            String[] parts = ipAddress.split("\.");
            int firstOctet = Integer.parseInt(parts[0]);
            String ipClass = "";
            if (firstOctet >= 0 && firstOctet <= 127)
                ipClass = "Class A";
            else if (firstOctet <= 191)
                ipClass = "Class B";
            else if (firstOctet <= 223)
                ipClass = "Class C";
            else if (firstOctet <= 239)
                ipClass = "Class D (Multicast)";
            else
                ipClass = "Class E (Experimental)";
            System.out.println("IP Address Class: " + ipClass);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

3. Java Client-Server Program for Uppercase Conversion

```
Server (UppercaseServer.java):
------------------------------
import java.io.*;
import java.net.*;

public class UppercaseServer {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(5001);
        Socket clientSocket = serverSocket.accept();
                                  BufferedReader    in    =    new    BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
        String input = in.readLine();
        String output = input.toUpperCase();
        out.println(output);
        clientSocket.close();
        serverSocket.close();
    }
}

Client (UppercaseClient.java):
------------------------------
```

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

public class UppercaseClient {
    public static void main(String[] args) throws IOException {
        Socket socket = new Socket("localhost", 5001);
        BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
                                  BufferedReader    in    =    new    BufferedReader(new
InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        System.out.print("Enter string: ");
        String input = userInput.readLine();
        out.println(input);
        String response = in.readLine();
        System.out.println("Server returned: " + response);
        socket.close();
    }
}
```