Computer Networks

Lab Report-8

NAME : **Ammar Younas**

COURSE: **BESE-26C**

EXERCISES

Exercise 8.1: **[5]**

Write a client Server Application in which, clients sends four digit numbers i.e., “3145” to server. Server sends back the sum of all digits to client i.e., 13. Use Socket and port of your choice.

CLIENT:

```
import java.io.*;
import java.net.*;
class SumClient{
public static void main(String args[]) throws Exception
{
    String num;
    int result;
    System.out.print("Enter a number: ");
    BufferedReader inFromUser=new BufferedReader(new
    InputStreamReader(System.in));
    try (Socket clientSocket = new Socket("localhost", 6770)) {
    DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
    BufferedReader inFromServer = new BufferedReader(new
    InputStreamReader(clientSocket.getInputStream()));
    num = inFromUser.readLine();
    outToServer.writeBytes(num+ '\n');
    result = inFromServer.read();
    System.out.println("The response from server is: " + result);
    }
    }
}
```

SERVER:

```
import java.io.*;
import java.net.*;
class SumServer{
public static void main(String args[]) throws Exception
{
    String num;
    int sum=0;
    ServerSocket welcomeSocket = new ServerSocket(6770);
    while(true) {
        Socket connectionSocket = welcomeSocket.accept();
        BufferedReader inFromClient = new BufferedReader(new
        InputStreamReader(connectionSocket.getInputStream()));
        DataOutputStream outToClient = new
DataOutputStream(connectionSocket.getOutputStream());
        num = inFromClient.readLine();

        for (int i=0;i!=num.length();i++){
```

```
            sum+=Integer.parseInt(num, i);
            }
    outToClient.write(sum);
    }
    }
    }
```

OUTPUT:

```
PS C:\Users\Ammar Younas\Desktop\CN Codes> cd
tput }
Enter a number: 3145
The Response from Server is: 13
```

Exercise 8.2: **[5]**

Write a program that will count the number of lines in file that is specified on the command line. Assume that the files are text files and placed in c:\ driver of server. Write file name, along with the number of lines in that file, to standard output at the client end. If an error occurs while trying to read from the files, you should print an error message.

CLIENT:

```
import java.io.*;
import java.net.*;
class FileClient{
public static void main(String args[]) throws Exception
{
    int i;
    try (Socket clientSocket = new Socket("localhost", 6771)) {
    File file = new File("C:\\ammar.txt");
    FileInputStream fileInputStream = new FileInputStream(file);
    InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream);
    BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
    DataOutputStream outToServer =new DataOutputStream(clientSocket.getOutputStream());
    BufferedReader inFromServer = new BufferedReader(new
    InputStreamReader(clientSocket.getInputStream()));
    while ((bufferedReader.readLine()) != null) {
    outToServer.writeBytes(bufferedReader.readLine());
    }
    i= inFromServer.read();
    System.out.println(file.getName());
    System.out.println("The number of lines are: " + i);
    }
    }
}
```

SERVER:

```
import java.io.*;
import java.net.*;
class FileServer{
public static void main(String args[]) throws Exception
{
    int lines=0;
    ServerSocket welcomeSocket = new ServerSocket(6771);
    while(true) {
        Socket connectionSocket = welcomeSocket.accept();
        BufferedReader inFromClient = new BufferedReader(new
        InputStreamReader(connectionSocket.getInputStream()));
        DataOutputStream outToClient = new
DataOutputStream(connectionSocket.getOutputStream());
        while ((inFromClient.readLine()) != null) {
            inFromClient.readLine();
            lines++;
            }
        outToClient.write(lines);
    }
}
}
```

OUTPUT:

```
The Response from Server is: 13
PS C:\Users\Ammar Younas\Desktop\CN Codes
tput }
ammar
The number of lines are: 3
```