EXPERIMENT 7

Aim :

Implement simple Multithreaded Server to perform all Mathematics Operations Parallel in Java.

Code :

Server Class : Server.java file

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

//Server class
class Server {
    public static void main(String[] args)
    {
        ServerSocket server = null;

        try {

            //server is listening on port 1234
            server = new ServerSocket(1234);
            server.setReuseAddress(true);

            //running infinite loop for getting client request
            while (true) {

                //socket object to receive incoming client requests
                Socket client = server.accept();

                //displaying that new client is connected to server
                System.out.println("New client connected"
                                   + client.getInetAddress()
                                           .getHostAddress());

                //create a new thread object
                ClientHandler clientSock
                    = new ClientHandler(client);

                //this thread will handle the client separately
                new Thread(clientSock).start();
            }
        }
        catch (IOException e) {
            e.printStackTrace();
        }
```

```
        finally {
            if (server != null) {
                try {
                    server.close();
                }
                catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    //ClientHandler class
    private static class ClientHandler implements Runnable {
        private final Socket clientSocket;

        //Constructor
        public ClientHandler(Socket socket)
        {
            this.clientSocket = socket;
        }

        public void run()
        {
            PrintWriter out = null;
            BufferedReader in = null;
            try {

                //get the outputstream of client
                out = new PrintWriter(
                    clientSocket.getOutputStream(), true);

                //get the inputstream of client
                in = new BufferedReader(
                    new InputStreamReader(
                        clientSocket.getInputStream()));

                String line;

                while ((line = in.readLine()) != null) {
                    String[] arr=line.split(" ");
                    int res=0,p,q;
                    p=Integer.parseInt(arr[1]);q=Integer.parseInt(arr[2]);

                    if(line.charAt(0)=='1') {
                        res=p+q;
                    }
```

```
                    if(line.charAt(0)=='2') {
                        res=p-q;
                    }
                    if(line.charAt(0)=='3') {
                        res=p*q;
                    }
                    if(line.charAt(0)=='4') {
                        res=p/q;
                    }

                    String text="choice: "+String.valueOf(line.charAt(0))+
"\nAnd the Numbers are "+p+" "+q;

                    //writing the received message from client
                    System.out.printf(
                        " Sent from the client: %s\n", text);

                    line=Integer.toString(res);
                    out.println(line);
                }
            }
            catch (IOException e) {
                e.printStackTrace();
            }

            finally {
                try {
                    if (out != null) {
                        out.close();
                    }
                    if (in != null) {
                        in.close();
                        clientSocket.close();
                    }
                }
                catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

Code :

Client Class : Client.java file

```
import java.io.*;
import java.net.*;
import java.util.*;
//Client class
class Client {
    //driver code
    public static void main(String[] args) {
        //establish a connection by providing host and port number
        try (Socket socket = new Socket("localhost", 1234)) {
            //writing to server
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
            //reading from server
      BufferedReader in = new BufferedReader(new InputStreamReader(
                    socket.getInputStream()));
            //object of scanner class
            Scanner sc = new Scanner(System.in);
            String line = null;
            while (!"exit".equalsIgnoreCase(line)) {
                //reading from user
                System.out.println("\nEnter Numbers: ");
                String a1,a2;
                a1=sc.nextLine();
                a2=sc.nextLine();
                System.out.println("1. Addition\n2. Subtraction\n3. Multip
lication\n4. Division");
                String ch;
                System.out.println("\nEnter Choice: ");
                ch=sc.nextLine();
                line=ch+' '+a1+' '+a2;
                //sending the user input to server
                out.println(line);
                out.flush();
                //displaying server reply
       System.out.println("Server replied Result is: " + in.readLine());
}
            //closing the scanner object
            sc.close();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output Screenshots :